Department of Biotechnology and Chemical Technology

Networked Control with Delay Measurement and Estimation

Nikolai Vatanski

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 100/2011

Networked Control with Delay Measurement and Estimation

Nikolai Vatanski

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the School of Chemical Technology for public examination and debate in Auditorium KE2 (Komppa Auditorium) at Aalto University School of Chemical Technology (Espoo, Finland) on the 18th of November, 2011, at 12 noon.

Aalto University
School of Chemical Technology
Department of Biotechnology and Chemical Technology

Supervisor
Professor Sirkka-Liisa Jämsä-Jounela

Instructor
Professor Raimo Ylinen

Preliminary examiners
Professor Dominique Sauter, Nancy Poincaré University, France
Professor Matti Vilkko, Tampere University of Technology, Finland

Opponents
Professor Dominique Sauter, Nancy Poincaré University, France
Professor Steven X. Ding, University of Duisburg-Essen, Germany

Aalto University publication series
DOCTORAL DISSERTATIONS 100/2011

ISBN 978-952-60-4325-8 (pdf)
ISBN 978-952-60-4324-1 (printed)
ISSN-L 1799-4934
ISSN 1799-4942 (pdf)
ISSN 1799-4934 (printed)

Unigrafia Oy
Helsinki 2011

Finland

The dissertation can be read at http://lib.tkk.fi/Diss/

A! Aalto University

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Nikolai Vatanski

Name of the doctoral dissertation
Networked Control with Delay Measurement and Estimation

Publisher School of Chemical Technology

Unit Department of Biotechnology and Chemical Technology

Series Aalto University publication series DOCTORAL DISSERTATIONS 100/2011

Field of research Process Control

Manuscript submitted 18 April 2011 **Manuscript revised** 12 August 2011

Date of the defence 18 November 2011 **Language** English

☒ **Monograph** ☐ **Article dissertation (summary + original articles)**

Abstract

As future industrial communication network solutions move towards industrial Ethernet based solutions, monitoring and control of network parameters and control of the network will gain more attention than previously in order to achieve unperturbed performance. The main challenges in implementation of Ethernet based solutions in control applications come from the fact that the performance of the network is priory nondeterministic and may change according to the load in the network. Respectively, this perturbed operation of the communication network may affect on efficiency and performance of the control application.

This work focuses on performance improvement of a control application that is controlled over a switched Ethernet network. In such networks main network parameter that may pose perturbation in performance is the network-induced delay. With novel network instrumentation and network performance estimation methods the behavior of the network-induced delay in the switched Ethernet network is described. The outputs of these methods are used in the control strategies to compensate for the deterious effect of the network.

First, a method for measuring the network information, particularly the network-induced transmission delay, is presented. Next, this information is utilized in control design. Two control approaches are developed, which differ with respect to the time-variant behavior of the delay measurements. A Smith predictor based approach is proposed for the case where network time delay variation is low, and polynomial system based time variant controller and observer design is proposed for the case where time variation is high. Second, methods for estimating the network-induced transmission delay, is presented and utilized in control design. Two control approaches based on robust control theory are developed, which differ with respect to ability to take into account in addition of delay variation also other uncertainty sources, such as model uncertainty and measurement errors.

The developed control designs are tested with simulations as well as on the experimental platform. From the results it is clear that utilization of the methods improve performance of the application. From the results it is also clear that no single method is suitable for all the cases, but the choice should be based on detail investigation of available instrumentation, magnitude of network delay variation, and uncertainty sources coming from the process.

Keywords Networked control systems, delay compensation, switched Ethernet, polynomial systems theory

ISBN (printed) 978-952-60-4324-1 **ISBN (pdf)** 978-952-60-4325-8

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2011

Pages 168 **The dissertation can be read at** http://lib.tkk.fi/Diss/

A! **Aalto-yliopisto**

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tiivistelmä

Tekijä
Nikolai Vatanski

Väitöskirjan nimi
Viiveen mittaamiseen ja estimointiin perustuva verkottuneiden järjestelmien ohjaus

Julkaisija Kemian tekniikan korkeakoulu

Yksikkö Biotekniikan ja kemian tekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 100/2011

Tutkimusala Prosessien ohjaus

Käsikirjoituksen pvm 18.04.2011 **Korjatun käsikirjoituksen pvm** 12.08.2011

Väitöspäivä 18.11.2011 **Kieli** Englanti

☒ **Monografia** ☐ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Teollisuuden tietoliikenneverkkojen kehittyessä kohti teollisuus-Ethernet pohjaisia järjestelmiä on häiriöttömän toiminnan varmistamiseksi kiinnitettävä huomiota verkon ohjaukseen ja verkon parametrien monitorointiin ja estimointiin. Eräs päähaasteista Ethernet-pohjaisten ratkaisujen käytössä prosessien ohjaussovelluksissa on verkon suorituskyvyn epädeterministisyys eli sen suorituskyvyn muuttuminen kuormituksen mukaan. Verkon häiriöllinen toiminta voi edelleen vaikuttaa ohjaussovelluksen suorituskykyyn ja tehokkuuteen.

Tässä työssä keskitytään sellaisen säätösovelluksen suorituskyvyn parantamiseen, joka on rakennettu Ethernet-verkkoa hyödyntäen. Tällaisissa verkoissa parametri, joka voi aiheuttaa häiriöitä säätösovelluksen toimintaan, on tietoliikenteestä johtuva viive. Uusilla verkon instrumentointimenetelmillä ja toimintaparametrien estimointimenetelmillä tietoliikenteestä aiheutuva viive voidaan arvioida ja käyttää tuloksia säätösuunnittelussa verkosta johtuvien häiriöiden kompensoimiseen.

Tässä työssä esitetään aluksi menetelmä, jolla voidaan mitata verkosta johtuvaa viivettä. Tätä informaatiota käytetään kahdessa eri säätöstrategiassa, joilla on erilainen kyky ottaa huomioon viiveen vaihtelun suuruutta säädössä. Smith-prediktoripohjaista ratkaisua esitetään käytettäväksi tapauksissa, joissa verkon viiveen vaihtelu on pieni. Polynomisysteemipohjaista aikavarianttia säädintä ja -havaitsijaa ehdotetaan käytettäväksi vastaavasti tapaukseen, jossa viiveen vaihtelu on suuri. Seuraavaksi työssä esitetään menetelmä, jolla voidaan estimoida verkosta johtuvaa viivettä, ja tätä tietoa käytetään edelleen hyväksi säätösuunnittelussa. Työssä kuvataan kaksi robustiin säätöteoriaan pohjautuvaa menetelmää, jotka eroavat toisistaan siinä, kuinka hyvin ne huomioivat verkon viiveen vaihtelun lisäksi myös muita epävarmuustekijöitä, kuten malliepävarmuutta ja mittausvirhettä.

Kehitettyjä säätöstrategioita testataan simuloinnin lisäksi myös Ethernet testausalustalla. Tuloksista ilmenee, että käyttämällä näitä menetelmiä voidaan parantaa huomattavasti sovelluksen toimintaa, ja täten vähentää verkon aiheuttamaa haitallista vaikutusta säätösysteemiin. Tulosten perusteella voidaan myös todeta, ettei mikään yksittäinen menetelmä sovellu kaikkiin tapauksiin, vaan menetelmän valinnan tulisi perustua saatavissa olevaan instrumentointiin, viivevaihtelun suuruuteen ja prosessista tulevien epävarmuustekijöiden huomioimiseen.

Avainsanat Verkottuneet ohjausjärjestelmät, viiveen kompensointi, kytketty Ethernet, polynomisysteemiteoria

ISBN (painettu) 978-952-60-4324-1 **ISBN (pdf)** 978-952-60-4325-8

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2011

Sivumäärä 168 **Luettavissa verkossa osoitteessa** http://lib.tkk.fi/Diss/

Preface

The research work presented in this thesis has been carried out in the Laboratory of Process Control and Automation, Helsinki University of Technology, mainly between May 2004 and November 2008. I would like to thank my supervisor Professor Sirkka-Liisa Jämsä-Jounela for encouraging me to start PhD studies, for her comprehensive academic guidance during the years and for her help and support in the final stages of the thesis. I am greatly indebted to Professor Raimo Ylinen for introducing me to the fundamental ideas behind polynomial systems theory and for all his valuable advices and discussions.

I am very grateful to Professor Eric Rondeau for welcoming me into his research group in Henri Poincaré University. I wish to express my gratitude for the group for the great atmosphere during the unforgettable stay in France, with special thanks to Dr. Jean-Philippe Georges for creative discussions. I would like to thank the pre-examiners of this thesis, Professor Dominique Sauter and Professor Matti Vilkko for their review and for their comments. I thank Finnish Culture Foundation, Research Foundation of Helsinki University of Technology, Walter Ahlström foundation, and Outotec Oyj for additional funding.

I would like to thank all my former colleagues in the laboratory for friendly atmosphere. Especially, Jerri Kämpe, Cheng Hui, Mikko Vermasvuori, Mats Nikus, Alexey Zakharov, Antti Remes, Tiina Komulainen, Ville Suontaka and Markus Kettunen, for keeping up the spirits.

I would like to express my deepest gratitude to my parents, Viktor and Sylvi, and my siblings, Victoria and Aleksander, for all their support and dedication over the years. Finally, I would like to express my deepest gratitude to Mirka, for her love and support.

Espoo, September 21, 2011

Nikolai Vatanski

Abbreviations

ACK	-	Acknowledgement signal
ARINC	-	Aeronautical Radio Incorporated
BIBO	-	Bounded Input Bounded Output
CAN	-	Controller Area Network
CSMA/CD	-	Carrier Sense Multiple Access with Collision detection
CUT	-	Canonical Upper Triangular form
DCS	-	Digital Control System
DMC	-	Dynamic Matrix Control
ERP	-	Enterprise resource planning
FF	-	Foundation Fieldbus
FIFO	-	First In First Out
FIP	-	Factory Instrumentation Protocol
FIR	-	Finite Impulse Response
GPC	-	Generalized Predictive Control
IAONA	-	Industrial Automation Open Networking Alliance
IAE	-	Integral of Absolute Error
ICT	-	Information and Communication Technology
IEC	-	International Electrotechnical Commission
IEEE	-	Institute of Electrical and Electronics Engineers
IP	-	Internet Protocol
ISA	-	International Society of Automation
ISE	-	Integral of Squared Errors
ISO	-	International Standards Organization
I/O	-	Input/Output
IO	-	Input Output
LAN	-	Local Area Network
LBD	-	Lower Bound Delay
LFT	-	Linear Fractional Transformation

LMI	-	Linear Matrix Inequality
LQG	-	Linear-Quadratic-Gaussian
LQR	-	Linear-Quadratic Regulator
LLC	-	Logical Link Control
MAC	-	Medium Access Protocol
MES	-	Manufacture Execution System
MADB	-	Maximum Allowable Delay Bound
MATI	-	Maximum Allowable Transfer Interval
MAC	-	Medium Access Control
MIMO	-	Multiple Input Multiple Output
MPC	-	Model Predictive Control
NCS	-	Networked Control System
NTP	-	Network Time Protocol
OSI	-	Open Systems Interconnection
PI	-	Proportional-Integral
PD	-	Proportional-Derivative
PID	-	Proportional-Integral-Derivative
PLC	-	Programmable Logic Controller
PTP	-	Precision Time Protocol
RP	-	Robust Performance
RM	-	Rate Monitoring
RFDE	-	Retarder Functional Differential Equation
RT	-	Real Time
RTAI	-	RealTime Application Interface
RTE	-	Real Time Ethernet
RTT	-	Round Trip Time
SISO	-	Single Input Single Output
QoS	-	Quality of Service
TCP	-	Transmission Control Protocol
TDC	-	Time Delay Compensator
UBD	-	Upped Bound Delay
UDP	-	User Datagram Protocol
WLAN	-	Wireless Local Area Network
WorldFIP	-	World Factory Instrumentation Protocol
ZOH	-	Zero-order-hold

List of Symbols

GREEK SYMBOLS

α	-	Stochastic parameter for loss modeling
Δ	-	Perturbation in uncertainty representation
$\bar{\delta}$	-	Maximum delay error
ε	-	Error signal, discrete case
Φ	-	State matrix, discrete state space representation
Γ	-	Input matrix, discrete state space representation
λ	-	Eigenvalue
μ	-	Structured singular value
μ_i	-	Admissible control policy
$\vartheta(.)$	-	Time varying delay represented with unit prediction operator
θ	-	Delay value
$\theta(.)$	-	Time varying delay represented with unit delay operator
$\theta_{\min}$	-	Minimum value of the delay
$\theta_{\max}$	-	Maximum value of the delay
ρ	-	Traffic flow
σ	-	Maximum amount of data that can arrive in a burst (burstiness)
τ	-	Overall network transmission delay
$\dot{\tau}$	-	Variation rate of network transmission delay
τ_i^j	-	Network transmission delay at *i* with the value *j*
τ_{bw}	-	Delay due to the limited bandwidth
τ_c	-	Control induced delay
τ_{ca}	-	Controller to actuator transmission delay
$\hat{\tau}_{ca}$	-	Measurement value for controller to actuator transmission delay
$\tau_{ca,n}$	-	Controller to actuator transmission delay for actuator *n*
τ_{comp}	-	Computation delay at the controller node

τ_{nw}	-	Network transmission delay
τ_{oh}	-	Delay due to overhead in the communicating nodes
$\tau_{\max}$	-	Maximum value of the time constant
$\tau_{\min}$	-	Minimum value of the time constant
τ_{sc}	-	Sensor to controller transmission delay
$\hat{\tau}_{sc}$	-	Measurement value for sensor to controller transmission delay
$\tau_{sc,m}$	-	Sensor to controller transmission delay for sensor m
τ_{tv}	-	Time-varying network transmission delay
ω	-	Angle frequency
ω_B	-	Closed loop bandwidth

ROMAN SYMBOLS

A	-	State matrix in state space representation
B	-	Input matrix in state space representation
$\overline{B}$	-	Backlog upper bound value
C	-	Output matrix, in state space representation
C_{in}	-	Capacity of the input port
C_{out}	-	Capacity of the output port
D	-	Feedforward matrix in state space representation
$\overline{D}$	-	Transmission delay upper bound value
e	-	Error signal
E_c	-	Error signal reaching the conroller
F	-	Controller gain, discrete case
G	-	Plant transfer function
G_c	-	Controller transfer function
$\hat{G}$	-	Plant model
h	-	Sampling period
I	-	Identity matrix
I_i	-	Information vector
k	-	Index of time
K	-	Controller gain, continuous case
L	-	Loop transfer function
N	-	Decision horizon

M	-	Desired maximum peak of sensitivity function
q	-	Unit prediction operator
R	-	Reference signal
r	-	Unit delay operator
P	-	Plant model in discrete time
p	-	Differentiation operator
s	-	Laplace transform variable
S	-	Sensitivity function, IO-relation
S_I	-	Internal IO-relation
S_o	-	Overall IO-relation
S_{obs}	-	IO-relation for obeserver
T	-	Complementary sensitivity function
T_i	-	Transaction i of PTP
u	-	Input, input set
v	-	Stochastic Gaussian white noise
V	-	Lyapunov function
w	-	Stochastic Gaussian white noise
w_P	-	Weight for the sensitivity function
w_I	-	Weight for the complementary sensitivity function
x	-	State vector, input set
x_e	-	Equilibrium point
y	-	Output, output set
z	-	Z-transform complex variable
z_i	-	Exogenous output i

MATHEMATICAL NOTATIONS

$\overset{\Delta}{=}$	-	Equals by definition
$\subset$	-	Is a subset of
$\in$	-	Is an element of
$\Rightarrow$	-	Implies
$[A(r) \;\vdots\; -B(r)]$	-	Generator for difference input-output relation, expressed with the unit delay operator
$[A(q) \;\vdots\; -B(q)]$	-	Generator for difference input-output relation, expressed with the unit prediction operator
$\det(.)$	-	Determinant of $(.)$

$f : A \to B$	-	f is a mapping from the set A to the set B
H_2	-	H_2 norm
H_∞	-	H-infinity norm
$\ker(.)$	-	Kernel of $(.)$
r_K	-	Unit delay operator on the coefficient space K
$\mathbb{R}$	-	Field of real numbers
$\mathbb{Z}$	-	Ring of integer numbers
$\mathbb{C}$	-	Field of complex numbers
0_K	-	Zero operator on the coefficient space K

List of Tables

List of Figures

Contents

1 Introduction

Process automation systems of the future and even those currently in use today, will consist of a large number of intelligent devices and control systems connected by local or global communication networks. The intelligent features, some of which are directly embedded in the devices themselves, include maintenance or monitoring functions, advanced control algorithms, and machine vision functions. The addition of intelligent features to devices, in turn, increase performance requirements for the networks as more data needs to be transferred while satisfying real time requirements posed by the feedback control of the application.

For the control of industrial applications with real time requirements, industrial network systems - referred to as fieldbuses - were developed in the 1990s. Several competing technologies for fieldbuses are now in use with different performance characteristic related to real time control and the amount of data that can be transferred. From an implementation point of view, as the systems are very different, process device manufacturers are forced to implement support for each fieldbus separately, which increases hardware and software costs while adding uncertainty to interoperability. The original intention for one single unified communications mechanism has not been realized. As the latest trend, Ethernet-based technologies have attracted attention because of its simplicity, cost and wide acceptance as an alternative to the fieldbus. However, it has been known that the standard Ethernet is not directly suitable for industrial networking due to the non-deterministic performance of the medium access control method that exhibits unstable performance under heavy traffic and unbounded delay distribution. Recently, a modification of the standard version, the switched Ethernet, has shown a very promising prospect for real-time industrial networking because the switching technology using full duplex mode can eliminate frame collisions common in a standard Ethernet. This removes non-deterministic effects coming from the loss of data, to the congestion problem of the network switches introducing an additional delay. As future industrial communication solutions move to an increasing extent towards the industrial Ethernet solutions, it will be necessary to consider the monitoring and control of the network parameters in order to achieve an undisturbed performance. The network transmission delay, in particular, will be the key performance parameter describing the behavior of the network that needs to be considered in the control system design.

Research that explores the feedback control of an application with high real-time requirements under varying network conditions is research for networked control systems (NCS). Traditionally, networked control systems are defined as feedback control systems wherein the control loops are closed through a real-time network (Wang and Liu, 2008). In studies on the NCS, the communication network is considered to be an important part of the control loop, the performance of which significantly influences the overall performance of the controlled application. To mitigate network effects, special modeling approaches, control algorithms, network adjustment methods and analysis approaches are developed, which adopt some of the principles of control and network theories.

The field of networked control systems covers two closely related topics: control over network and control of network. In control over network, the network is considered as a passive resource that has constraints expressed in terms of certain network performance measures, where the goal is to design a control algorithm that is able to adapt to network perturbations to guarantee the performance of the application. In control of network, the network is an active resource and can be controlled. The goal is to optimize network parameters such that adequate network performance is guaranteed for the performance of the application. A combination of these where the performances of the control and the network performance are optimized simultaneously is sometimes referred as a co-design approach (integrated approach). A control method for NCS can thus have two meanings: the method can be either a control over network method or control of network method. As this work mainly concentrates on control over network, when discussing a control method for NCS, the control over method is generally meant. When discussing the control of network method for NCS, this will be explicitly mentioned in the context.

Interest in the study of networked control systems has been intense over the last decade. The topic is still very timely in modern control theory engineering, and is in its relative infancy compared to many other control theory fields, both in terms of theory and industrial applications. A number of review papers have been published in the field of NCS, including a general collection of the main research directions in the handbook by Hristu-Varsakelis and Levine (2005); a first survey for NCS by Tipsuwan and Chow (2003); and a comprehensive update to the previous by Yang (2006). In the field of networked control systems, the control methods are usually dependent on how the network has been represented. As the topic is not very well established, a large number of modeling methods for the network has been presented, which means that the proposed control methods are also rather diverse. In case the network is modeled as a time varying delay block, a variety of methods from control theory have been proposed.

One of the first highly referenced pieces of research is by Nilsson (1998), where the network delay variation has been modeled using a Markov chain, which gave a Markovian jump linear closed-loop system with two modes as a NCS model. Nilsson (1998) proposed a Linear-Quadratic- Gaussian (LQG) controller to solve the problem; following Nilsson, several other researchers, such as Shousong and Qixin (2003), extended the result by taking into account longer delays and packet dropouts. In the case where the network has been modeled with sampled data models, H_∞ norm optimization based approaches have been commonly utilized to solve the control problem. Basic controllers, such as PID or MPC controllers have been used particularly used in connection with experimental studies.

Research on the stability of NCS has also been very active. Stability results from the field of nonlinear control theory, such as the Lyapunov second method, or the time variant control theory, such as the Krasovsky-Lyapunov method, have also been applied. New terminology that has been proposed and used in connection with research on stability of NCS is maximum allowable transfer interval (MATI) and maximum allowable deadband (MADB). MATI is a maximum value for transmission delay and, as long the transmission delay remains under this value, the system is considered stable. Several research groups have proposed their methods in order to determine MATI. MADB describes the delay band for sampling for which stability can be guaranteed. Research has been carried out to find an appropriate network scheduling method that limits the network-induced delay to less than the MADB. In addition to theoretical research where the network has been represented as a delay, some research has been carried out in connection with available fieldbuses. In Lian, *et al.* (2001), for example, a comparative analysis of three types of networks is presented - Ethernet, ControlNet and DeviceNet - where their performance has been evaluated with the respect of requirements brought by the control. For the switched Ethernet network, Lee, *et al.* (2006) studied the temporal effects of the switched Ethernet on control applications. In summary, most of the current research for networked control systems has been analytically oriented; further, there is no good example of theoretical development based on a specific industrial application (Yang, 2006). In order to obtain methods that could be also applied in an industrial environment, more research should be carried out in connection with existing communication networks.

As future industrial communication network solutions move towards industrial Ethernet based solutions, the monitoring and the control of the network parameters and network will gain more attention than previously in order to achieve unperturbed performance. The key factors in achieving this goal are the parameters describing the network performance, the estimation and control of those. The major research problem and motivation for this doctoral thesis is the improvement of the control performance of an application that is controlled over the switched Ethernet network. An improvement in performance is gained through the utilization of network oriented models, or measurements from the network, which are able to describe the time-delay effect introduced by the network. As the performance of the switched Ethernet is better known, it enables the utilization of dedicated control methods which improve the performance of the controlled application, increases the reliability of the controller, and decreases the malfunctions of the process which, over time, improve the profitability of the target application.

1.1 Research problem and asserted hypotheses

The focus in this thesis is on development of control algorithms for applications that are controlled over a switched Ethernet network. It is assumed that the application is greatly affected by the network performance and therefore the key operation parameter in switched Ethernet, the network induced delay, will be utilized to improve performance. In this work, delay compensation methods for Networked Control Systems are proposed while considering the protocol specifications of the switched Ethernet. The proposed approaches will be classified into two categories depending on the information available from the network on its performance. In the first category, the measured information will be used to enhance the overall performance of the controlled application; in the second category, the estimated information obtained with a detailed network model is utilized for performance improvement.

For the approach with the use of measured information, two control compensation strategies are presented. These differ with respect to their applicability to the control of time-varying systems. The first takes the time variant behavior into account by adjusting the controller parameters based on the uncertainty bounds introduced by the time variance; in the second, time variant control methods are applied directly.

For the approach with the use of estimated information, an estimate for an upper bound delay in a switched Ethernet network will be obtained using architectural consideration of the network using network calculus-based approach as a tool. Two control compensation strategies are then presented that differ with the respect of uncertainty sources. In the first case, it is assumed that the only uncertainty stems from the network and process model, and that the process measurements are accurate. The second case also considers uncertainties emanating from the process, i.e. process model uncertainty and measurement errors. In both cases, robust control theory is used as a tool to generate robust controllers: in the first case, a controller is obtained by solving mixed H_∞ sensitivity problem; in the second, the controller is obtained via μ-synthesis using DK-iteration.

The main purpose of the proposed compensation strategies is to further reduce the non-deterministic effect of the switched Ethernet, thus making the protocol more suitable for the control applications. Through the increased use of Ethernet-based technology, the flexibility and modularity of the automation system improves thus reducing both maintenance and costs. In addition, the proposed methods will improve the operation of the local control loop which, in turn, will affect the performance of the advanced control systems as well as reducing costs and improving the efficiency of the production plant in general.

The asserted hypotheses of the thesis are:

1. With the network instrumentation and network performance estimation methods, it is possible to describe the behavior of the key performance parameters of the switched Ethernet network. The outputs of these methods enable the development of the control strategies to compensate for the deterious effect of the network.

2. The utilization of the estimated or the measured transmission delay information in control methods for NCS significantly increases the application performance and reduces the effects of non-deterministic properties of the switched Ethernet network.

In order to prove the hypotheses, the following tasks have been performed:

1. A method based on the time synchronization of devices or measuring end-to-end delay has been developed.
2. The obtained end-to-end delay information is utilized in control design. Two control approaches are developed, which differ with respect to the use of the time variant behavior of the network transmission delay. The Smith predictor-type method is used for the slow time varying case, and the polynomial system theory-based approach is used for the time varying case.
3. The obtained control designs are tested both with simulations as well as on an actual experimental prototype.
4. The network calculus based-method for estimating the upper bound of the network-induced transmission delay in a switched Ethernet network is presented.
5. The obtained information is utilized in control design. Two control approaches are developed which differ in respect to the uncertainty sources coming from the process. The H_∞-norm minimization-based method is used for the case with minor uncertainty in process measurements, and the μ-synthesis based-approach is proposed for a more general case.
6. The obtained control designs are tested both with simulations as well as on an actual experimental prototype.
7. The proposed control approaches are compared, and improved performance in terms of reduced uncertainty is indicated.

The first and fourth tasks increase the understanding of the network behavior enabling the development of the control design. In tasks two, three, five and six, the control design is performed. The increased overall performance and profitability of the operation is asserted within the seventh task.

1.2 Outline of the thesis

The aim of this thesis is to develop new control approaches for networked control systems that utilize the available estimated or measured information from the network. A switched Ethernet network is used as a network in NCS, for which the delay measurement and estimates for the delay are obtained.

In Chapter 2, the development of current industrial communication technologies are briefly presented. First, current industrial network architecture is presented. Next, an overview on the evolution of process layer communication networks is given, followed by a review on the next generation process layer networks and Ethernet based networks. The chapter concludes with an overview of a switched Ethernet as an improvement of a standard Ethernet, also illustrating the time variant effects that need to be considered.

Chapter 3 first presents a literature review of research work in the field of networked control systems (NCSs) before concluding with a summary of how control methods are classified with respect to the quality of information available from the NCS.

In Chapter 4, main aspects of the methods developed in this thesis are briefly described.

Chapter 5 presents a control approach for NCS that uses transmission delay measurements. First, a methodology to obtain the network delay measurement in a switched Ethernet network is presented and the obtained measurement is used in control compensation. As a compensation approach, a Smith predictor type delay observer is used and a PI controller is utilized as a control law. Time-variant behavior is taken into account by adjusting the controller parameters based on the uncertainty bounds introduced by the time variance. The control strategy is verified with the Matlab™ simulations as well as in the experimental platform developed for NCS analysis.

Where the control method for NCS is adopted from the time invariant control theory in Chapter 5, Chapter 6 utilizes a control method specifically designed for time variant control. In the chapter, the networked control system is modeled as a time-varying difference system, and the polynomial systems theory is applied for controller and observer design. In the design, it is particularly considered that the networked control system is a combination of time variant subsystems, which are shown to be non-commutative; therefore, algebraic methodology related to non-commutative skew polynomials is utilized.

Chapter 7 presents an improved control method for NCS using transmission delay estimations. In the chapter, a control strategy for the NCS using delay estimation is developed. First, a methodology to obtain the network transmission delay estimate in a switched Ethernet network is presented; the obtained estimate is then used in a stability analysis and control compensation. A control compensation strategy based on the robust control theory is proposed and further evaluated with the simulations at the end of the chapter.

In Chapter 8, a control approach for NCS is presented that considers not only time-varying effects induced by the network, but also uncertainty coming from inaccurate process measurements. In the chapter, a control problem is formulated using the tools from the robust control theory and a controller is formed using DK -iteration.

In Chapter 9, the presented control approaches are summarized. Finally, Chapter 10 concludes the thesis with a brief summary of the results and some suggestions for future work.

1.3 Contribution of the thesis

The main contributions and novelty presented in this thesis are:
1) The direct utilization of end-to-end network transmission delay measurement obtained from a switched Ethernet network in a predictor-based control algorithm.
2) The development and application of the time variant control methods based on the polynomial systems theory for networked control systems.
3) A new combination of the network theoretical methods (network calculus) with the control theory (robust control).
4) A new robust control design for delay and measurement error tolerance for the networked control system with a switched Ethernet as the network.

In contribution 1), the end-to-end delay measurement is obtained by synchronizing network devices using the IEEE 1588 protocol, and determining the transmission delay from time stamped packets. As a control algorithm, an adaptive Smith predictor is utilized. The adaptation rules are based on the uncertainty bounds introduced by the time variance. In contribution 2), the time variant polynomial systems theory is applied for the networked control system. In the development challenges introduced by the time variance are studied and handled, such as the non-commutativity of dynamical systems.

In contribution 3), a network calculus-based method is used to obtain the upper bound delay in a switched Ethernet network and the delay value is used in the design of a robust controller. The controller design is based on solving the mixed H_∞

problem. In contribution 4), the same network calculus-based estimate for the upper bound delay is utilized, but the controller is obtained via μ-synthesis. Resulted controller is robust also to other than network related uncertainties.

The original idea for determining end-to-end delay by time stamping has been presented at least by Jasperneite, *et al.* (2004). The adaptation rule based on the uncertainty bound has been presented in Morari and Zafiriou (1989). This principle has been implemented and modified by the author to suit the networked control system in order to improve performance of the application under the time variant delay.

The polynomial systems theory for time variant systems has been continuously developed by the Finnish research group of Blomberg and Ylinen since the 1970s. The principle has been applied for the networked control system by the author.

A model to obtain the upper bound end-to-end delay in a switched Ethernet network by architectural considerations of the network has been developed by Georges, *et al.* (2005). The robust control design method has been applied using Skogestad and Postlethwaite (2005) as a guide, the weights for the robust controller are from Wang, *et al.* (1994).

A robust control method that also takes other than network related uncertainties has been developed by applying Skogestad and Postlethwaite (2005), the weighting factors are developed by the author.

The work for contributions 1) and 3) is based on research work carried out by the author at the Research Centre for Automatic Control, Nancy, France, and the Laboratory of Process Control and Automation, Helsinki University of Technology (HUT). The research has been carried out in close cooperation with Dr. Jean-Philippe Georges. The author developed robust control based approaches that enabled the use of the estimate provided by the network calculus-based method, and a method to associate the classical theory tool with the QoS measurement method. The author was also involved in planning the control structure and the development of the experimental prototype and carried out the simulation experiments in cooperation with Dr. Jean-Philippe Georges. The results of this work are published in Vatanski, *et al.* (2009); however, the paper does not include the gain adaptation element and model uncertainty evaluations.

In the work for contribution 2), the author has developed the polynomial systems-based approach for the networked control under the supervision of Professor Raimo Ylinen, and carried out the simulation study. The results are published in Ylinen and Vatanski (2007) and Vatanski, *et al.* (2008) excluding consideration of the non-commutativitity and part of the simulations.

The work for contribution 4) has not been presented in any conference presentations or journals.

2 Industrial communication technologies

2.1 Current industrial networking architecture

The importance of industrial automation systems has increased dramatically in the recent decade. Automation systems are not anymore separated systems that handle only process control related tasks, but are interconnected with manufacturing execution and enterprise level systems to form a high level networking architecture. As a result, an entirely new industry sector has been formed and new automation solutions have a profound impact on the entire control philosophy of production plants.

The International Society of Automation defined during 2000-2005 an international ISA-95 standard for the integration of enterprise and control systems (ANSI/ISA-95.00.01-2000 2000). The standard manufacturing operations of an industrial plant are divided into hierarchical structure with four levels. The levels together with the high level network architecture are presented in Figure 1, where levels 0, 1 and 2 are the levels of process control. The objective is the control of a unit's processes and process equipment so that the end products with desired properties are obtained. Depending on the industry type, the control system may be either for batch, continuous type or discrete production processes. Level 0 is the level of actual physical production process; level 1 defines the activities involved in sensing the production process and manipulating the production process; and level 2 defines the activities of monitoring, supervisory control and automated control of the production process. Level 3 is the level of manufacturing execution system (MES). At this level several activities take place that must be executed to prepare, monitor and complete the production process at levels 0, 1 and 2. For example, activities like detailed scheduling, quality management, maintenance and production tracking take place at this level. The highest level (level 4) is called the level of Enterprise Resource Planning (ERP) systems. At this level, financial and logistic activities are executed that are not directly related to production, such as long term planning, marketing and sales, and procurement.

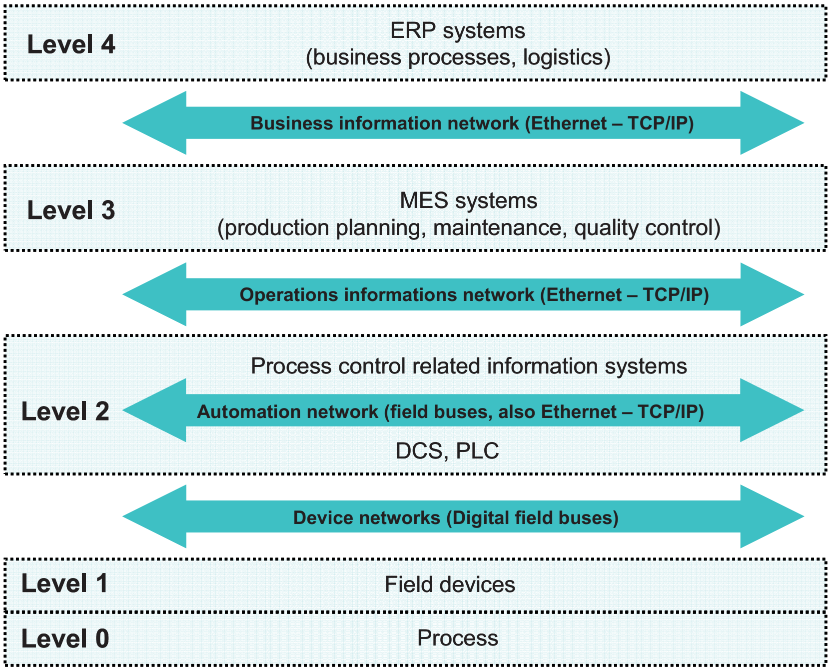

Figure 1. Current hierarchical communication structures as defined in the ISA 95 standard (ANSI/ISA-95.00.01-2000 2000).

From the networking perspective, the network used at level 4 is a business information network and Ethernet TCP/IP network is used at this level. The network used to interconnect levels 3 and 2 is the operations information network, also here the Ethernet is a common solution. Level 2 is the level of automation network; at this level, proprietary protocols have usually been used, but recently Ethernet-based solutions also are being utilized. The network between level 2 and 1 is called a device level network; here, fieldbus based networks are utilized.

The current trend in information and communication technology (ICT) and automation system development is a closer integration of ERP level functions (level 4), MES functions (level 3), and process control systems (levels 0, 1, and 2). The automation and MES systems, which have traditionally been integrated for production management, have been further developed to possess features of ERP systems. The ERP systems used for corporate-level resource planning and management are integrating new features that were previously used in MES and automation systems. The intelligence level is also increasing in the other direction, and basic control level devices like sensors that were traditionally used to obtain measurements, now have to support several maintenance or monitoring tasks. As the amount of information increases, and more information should be distributed across all levels in the hierarchy, from the information perspective, the traditional hierarchical structure is being replaced by a distributed architecture.

2.2 Evolution of process layer communication networks

As a result of the increased need for distributed architecture, more efforts have been put on communication technology development. The trend has been to move towards a single technology to achieve an open, modular, reliable and maintainable operation. The development of communication technology for communication networks used for the data exchange at the process layer is presented in the following.

The most common low-level industrial networks currently available can be classified into two main categories: traditional fieldbuses and Ethernet-based networks. Fieldbus networks are used at the device level where deterministic real time performance has to be achieved, while Ethernet-based networks are utilized at the upper level where strict real-time requirements can be relaxed.

The first traditional fieldbus standards were published in 1996 by the European CENELEC organization as an EN 50170 standard (CENELEC EN 50170, 1996). The standard initially consisted of the following four national standards: P-Net from Denmark (DS 21906, 1990), Profibus-FMS from Germany (DIN 19245-1:3, 1990), and WorldFIP from France (AFNOR NF C46-603, 1990). As each domain of the industry automation area has its specific needs and constraints, other fieldbuses have also been developed. For example, a fieldbus called Controller Area Network (CAN) has been developed for the automobile industry (Bosch, 1992; ISO 11898, 1993). The DeviceNet (Open DeviceNet Vendor Association, 1997), Interbus-S (DIN 19258-1, 1996), Lonworks (Echelon Corporation, 1995) and TTP/C (TTTech Computertechnik AG, 1996) are other examples of specific fieldbuses developed for a specific industry's automation domain. Furthermore, the EN 50170 standard (CENELEC EN 50170, 1996) was extended in order to define other networks: Foundation Fieldbus, ControlNet (ControlNet, 1997) and Profibus-PA (CENELEC EN 50170-2, 1996a). The technical committee has also published the EN 50254 standard (CENELEC EN 50254, 1996), Interbus (IEC 61784-2, 2007b), Profibus-DP (CENELEC EN 50170-2, 1996a), and Device WorldFIP, the EN 50325 (CENELEC EN 50325-4, 2002) (profiles derived from the CAN protocol), and the EN 50295 (CENELEC EN 50295, 1999) (AS-i protocol). The introduction of the fieldbuses based technology was a significant improvement for the operation of the automation systems. In addition to the improved communication, fieldbuses also enabled intelligent features such as self monitoring and fault diagnosis properties to be integrated closer to the process equipment, thus making it more intelligent.

One of the issues that held back (and still holds back) the use of fieldbuses in industrial practices was that the existing fieldbus technologies were largely vendor-specific, proprietary interfaces that preclude the ability to mix and match devices from different suppliers. In order to support a particular fieldbus, the device manufacturers have to implement functions required by a fieldbus separately; further, a certification fee must be paid in order to obtain official approval. This makes the cost of process equipment considerably higher. Also, several other interoperability problems occurred and so process equipment suppliers were unable to provide the devices for every fieldbus type. Due to these reasons and to the increased data transfer needs, automation systems have gradually favored the adoption of the currently popular office network technology – Ethernet-based networks - into industrial environments.

Several organizations (e.g. Industrial Automation Open Networking Alliance (IAONA)) are promoting the use of Ethernet in industrial automation. This has resulted that the current fieldbus standards also include Ethernet-based networks, which implement the Ethernet protocol in low layers such that time requirements of automation systems are better met. One example of this association is the Profinet network that combines Profibus and Ethernet technologies (CENELEC EN 50170-2, 1996b). Currently, the working group IEC WG11 is refining Real Time Ethernet (RTE) requirements to improve the profiles and the other related network components of international standards such as IEC 61784-1 (IEC 61784-1, 2007) and ISO/IEC 8802-3 (ISO/IEC 8802-3, 2000). Currently, the Ethernet-based industrial networks added to the IEC 61784 are Ethernet/IP; Profinet; Interbus; Vnet/IP (IEC 61784-2, 2007c); TCnet (IEC 61784-2, 2007d); EtherCAT (IEC 61784-2, 2007a); Powerlink (IEC 61784-2, 2007e); Modbus TCP (Transmission Control Protocol) (Modicon, 1999); and Sercos III (IEC 65C/358/NP, 2004).

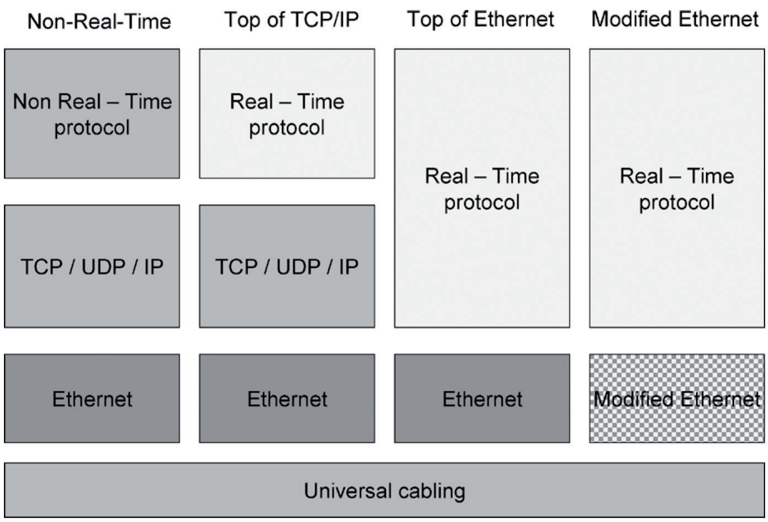

Figure 2. Possible structures for RTE (Felser, 2005).

These solutions try to modify the standard Ethernet by using three different approaches in principle. The first is to keep the TCP/UDP/IP protocols stack unchanged and concentrate all real-time modifications in the top layer. The handling of this communication protocol stack, however, requires reasonable resources in processing power and memory; however, nondeterministic delays in the communication still remain. Protocols that use this approach are Modbus/TCP, EtherNet/IP, ControlNet, DeviceNet, P-NET and Vnet/IP. In the second approach, the TCP/UDP/IP protocols are bypassed and the Ethernet functionality is accessed directly on top of the Ethernet. These RTE realizations do not alter the Ethernet communication hardware in any way, but are realized by specifying a special protocol type (Ethertype) in the Ethernet frame. Protocols that use this approach are Powerlink, TCnet and PROFINET CBA. In the third approach, the Ethernet mechanism and infrastructure itself is modified to make it more real-time (Modified Ethernet). Most solutions providing hard real-time services are based on modifications in the hardware of the device or the network infrastructure (switch or bridge). While the modifications are mandatory for all devices inside the network, they allow non-RTE traffic to be transmitted without modifications. Protocols that use this approach are SERCOS III, EtherCAT: EtherCAT and PROFINET IO. The difference between the approaches is shown in Figure 2.

2.3 Ethernet based network as a process layer network

Standard IEEE 802.3 (IEEE 802.3-2000, 2000), often referred as Ethernet, was developed for data communication among computers in the early 1970s by the

Institute of Electrical and Electronics Engineers (IEEE), and is the basis of physical and data link layers of the Internet for office communications. Ethernet was originally based on the idea of computers communicating over a shared coaxial cable acting as a broadcast transmission medium. From the early concept, Ethernet evolved into the complex networking technology that today underlies most local area networks (LANs).

Ethernet technology primarily exists at the data link layer of the ISO/OSI model architecture. The main goal of any data layer protocol (including Ethernet) is to provide mechanisms for framing, addressing and error detection. The data link layer is usually sub layered into two parts: the logical link control layer (LLC) and the medium access control (MAC) layer. The goal of the LLC layer is to provide either connectionless or connection-oriented data link service to the higher layer client, independent of the nature of the underlying LAN. The goal is that higher layer clients are relieved from having to manage with the details of the particular LAN technology being employed. The Medium Access Control (MAC) sub layer deals with the details of frame formats and channel arbitration associated with the particular LAN technology in use, independent of the class of service being provided to higher layer clients by LLC. The purpose of any MAC algorithm is to allow stations to decide when it is permissible for them to transmit on a shared physical channel. The main determining property of the Ethernet operation is that its MAC uses the Carrier Sense Multiple Access with Collision Detect (CSMA/CD) access method to handle simultaneous access demands.

The Carrier Sense Multiple Access with Collision Detect method is a set of rules determining how network devices respond when two devices attempt to use a data channel simultaneously. The main idea is that after detecting a collision, a device waits for a random delay time and then attempts to re-transmit the message. While this algorithm is simpler than those in the competing token ring or token bus technologies, it does, however, make the Ethernet non-deterministic in turns of frame transmission. When a station has a frame queued for transmission, it checks the physical channel to determine if it is currently in use by another station (carrier sensing). If the channel is busy, the station defers to the ongoing traffic to avoid corrupting a transmission in progress. Following the end of the transmission in progress (i.e., when the carrier is no longer sensed), the station waits for a period of time (an interframe gap) to allow the physical channel to stabilize and to additionally allow time for receivers to perform any necessary functions, such as adjusting the buffer pointers, updating management counters or interrupting a host processor, etc.

After the expiration of the interframe gap time, the station begins its transmission. If this is the only station on the network with a frame queued for transmission at this time, the station will be able to send its frame following the expiration of the interframe gap time with no interference from other stations. No other action is required and the frame is considered delivered by the sending station following the end of the frame transmission. However, if there are multiple stations with frames queued at the same time, each will attempt to transmit after the interframe gap expires following the deassertion of the carrier sense. The resulting interference is referred to as a collision. In the event of a collision, all involved stations continue to transmit for a short period to ensure that the collision is obvious to all parties. After jamming, the stations abort the remainder of their intended frames and wait for a random period of time. This is referred to as backing off. After backing off, the station goes back to the beginning of the process and attempts to send the frame again. If a frame encounters 16 transmission attempts all resulting in collisions, the frame is discarded by the MAC, the backoff range is reset and the event is reported to management. The simplified MAC flowchart of the transmission process of the Ethernet is provided in Figure 3.

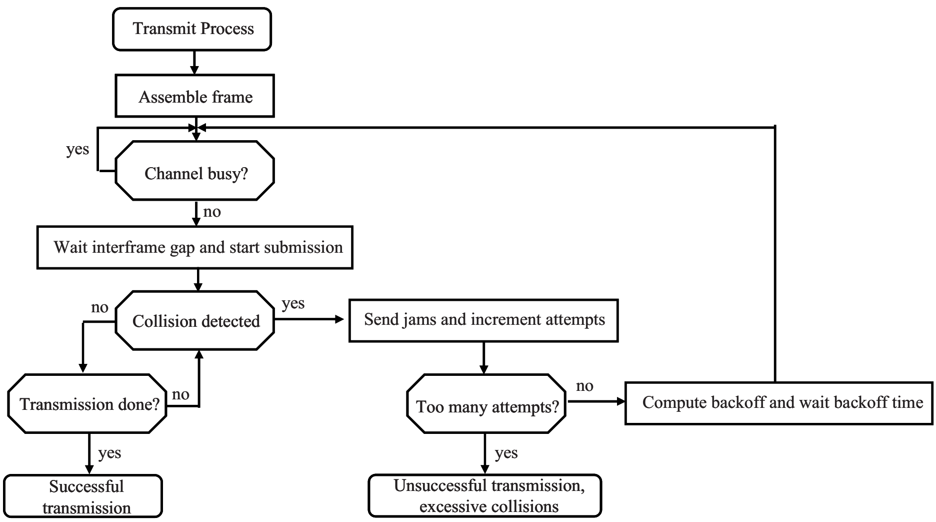

Figure 3. Ethernet MAC flow, transmit processes (Seifert, 2000).

2.4 Switched Ethernet as an improvement of the standard Ethernet

Physically, an Ethernet network can be constructed either by using hubs or switches. The respective names for the obtained network is the hubbed or switched Ethernet.

In the traditional Ethernet, the hub is a passive device that broadcasts whatever is received from source to other stations. The switch of the switched Ethernet is an active device that identifies the destination ports and relays the frame only to those. This means that if multiple stations are transmitting simultaneously, frames can be delivered without collision as long as their destinations are different from each other. An addition to this, the traditional Ethernet uses the half-duplex method, while the switched Ethernet uses the full-duplex method as the connection method between a station and the switch. This allows a station to transmit and receive frames simultaneously. The comparison of transmission methods of the traditional Ethernet and switched Ethernet is presented in Figure 4.

The differences between hubbed and switched Ethernet are important when developing the switched Ethernet automation system, since the switched Ethernet is free of frame collisions and thus there is no loss of data. However, even though at the increased network traffic the data is not lost, the delay to transfer data increases and this is deterious for the control application. A typical method of switching technology is the store and forward method as shown in the right hand side of Figure 4. In the figure, the switch receives a frame through a transmission line from a source station, and then checks if the reception line of a destination station is idle. If the reception line is idle, the switch transmits the frame. Otherwise, the switch stores the frame into its buffer and waits until the reception line becomes idle. If several frames with the same destination address are received at the switch simultaneously, the switch stores frames to the buffer and then sends frames to the destination one by one.

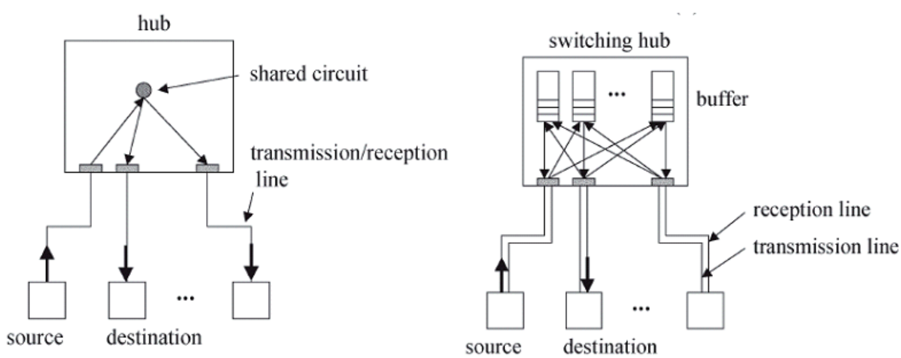

Figure 4. Comparison of transmission methods of traditional Ethernet and switched Ethernet, (Lee, *et al.*, 2006).

The time variant effect induced by the switched Ethernet is dependent on the load in the network. At low network loads, because of the low medium access overhead and the simplicity of the algorithm for operation, the switched Ethernet has almost no transmission delay at low network loads.

The situation is different at high network loads as the transmission delay over the Ethernet becomes uncertain. This is especially so for switched Ethernet networks, as the most common transient faults of the traditional Ethernet, collisions, are eliminated and shifted as a congestion problem in the switches. The transmission delay is often not constant, but varies according to the load in the network. The typical end-to-end transmission delay characteristic over a switched Ethernet network is presented in Figure 5. In the figure, the delay variation of 60,000 frames sent over an Ethernet switch is illustrated. Each frame had the size of 72 bytes, and the variation in delay was due to the background traffic generated by two other stations in the network. This additional transmission delay can be deterious for an application controlled over the network. In order to guarantee the applicability of the switched Ethernet for real-time control, the uncertainty of the delay magnitude and time variant behavior should be taken into account in the design.

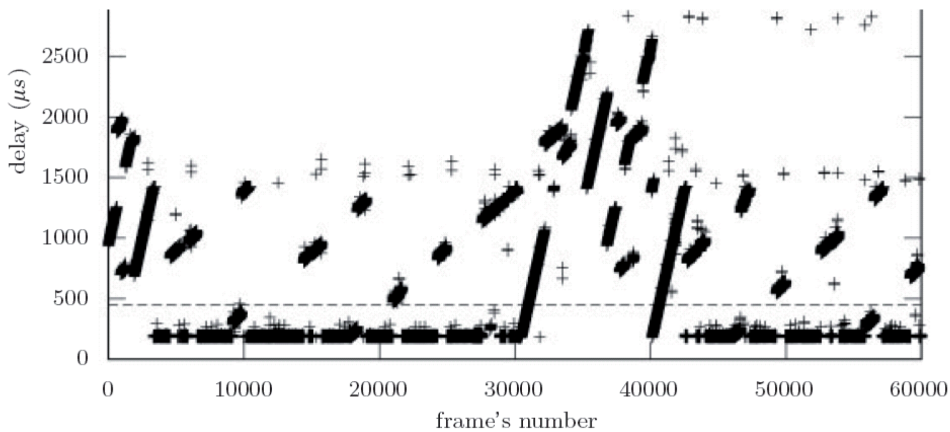

Figure 5. End-to-end delays in a switched Ethernet network, (Georges, *et al.*, 2005).

3 On networked control systems

Research for NCS is an interdisciplinary field combining control and network theories that study the systems in which the control loops are closed over a communication network. The communication network is considered an important part of the control loop, the performance of which significantly influences the overall performance of the controlled application. The network introduces several timing effects such as network delay, transient faults, jitter, asynchronization and sampling effects, the influence of which needs to be taken into account either by control algorithm, network control methods or by both. To mitigate these, special modeling approaches, control algorithms, network adjustment methods and analysis approaches have been proposed, which adopt some of the principles of control and network theories.

3.1 Structure of the networked control system

3.1.1 Feedback components and information flows

Physically, a typical NCS consists of a process, a controller, communication media as well as a number of sensors and actuators. Figure 6 shows a typical NCS block diagram and its information flows. This kind of setup is used when the network imposed timing effects are studied as a control problem. In the block diagram, the process G with m sensors and n actuators is controlled via communication media by the controllers G_c. The communication links from the sensor to the controllers and from the controllers to the actuators are illustrated with broken lines.

In the sensor nodes 1...m, the information on the state of the process is obtained, preprocessed and transmitted to the controller. The operation of the sensor node is usually time triggered and the operation period is denoted by the sampling period $h_{s,1},\ldots,h_{s,m}$. Signal processing, anti-aliasing filtering, sensor monitoring and similar services are usually hosted at the sensor node.

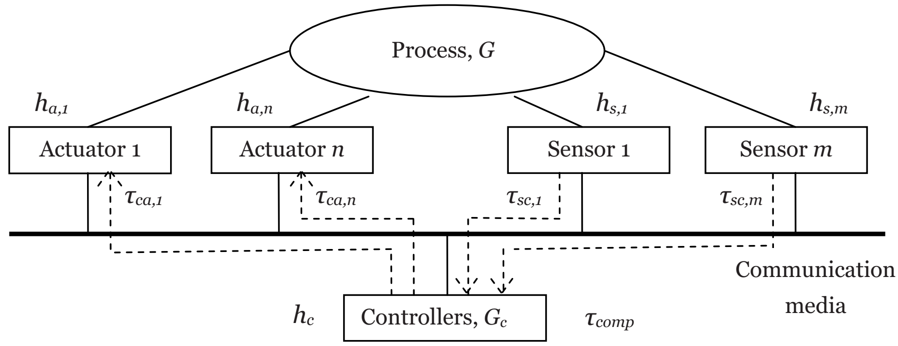

Figure 6. Typical setup and information flows in NCS.

In the controller node G_c, sensor information is used, the control action is calculated for every control period and submitted to the actuator. The controller can be time triggered or event triggered and the control period is often assumed to be equal to the sampling period of the sensor nodes $h_c = h_{s,i}$ for all i=1...m. The controller node often runs other tasks, which can be modeled as sources of delay and jitter. The overall delay induced by the controller node is presented as τ_{comp}.

In the actuator nodes 1...n, the control action is implemented. The actuator can be time or event triggered. It is rare to assume that the actuator node adds delay and jitter to the system, it is often assumed, that these are already inherited from the preceding tasks.

The communication links from the sensors to the controller can be time or event triggered. The communication can be managed through a dedicated link (direct link) or a link that operates according to some communication protocol. The links are often considered the source of delay (and delay variation), created either by contention in the communication or by transient faults. The transmission delay introduced by this link is depicted in Figure 6 as $\tau_{sc,1},...,\tau_{sc,m}$.

The communication links from the controller to the actuators is similar to the link from the sensors to the controller. The difference is that the time delay and jitter information lies in the future, as seen from a perspective of calculating the control law. The links introduce a delay that is depicted in Figure 6 as $\tau_{ca,1},...,\tau_{ca,n}$.

The performance of the NCS system is often assessed through the rise time, overshoot or similar time domain measures; the goal, therefore, is to design a control law so that the overall NCS performance is unaffected by delays and delay variation coming from the communication media.

3.1.2 Task sequence diagram

The task sequence diagram, that is more illustrative when the timing influence of the communication network in NCS is studied as a control of network problem, is presented in Figure 7. In the figure, five tasks are illustrated: the first task is related to the sensing action performed at the sensor node; the second is related to communication links from the sensor nodes to the controllers; the third is related to the calculation of the control action at the controller; the fourth is related to the communication from the controller node to the actuators; and finally the fifth task is related to the actions performed at the actuator node. This set of tasks form a task chain, which has to be scheduled together with other tasks not related to the control of the system. The execution of the task chain is usually periodic with the following task attributes: period, relative release time, relative deadline, worst case execution time and priority.

Whereas from the control engineering perspective the performance of the NCS can be assessed with the performance measures such as rise time and overshoot, from the communication network, the performance of the resulting system is measured in temporal properties by, for example, the number of missed deadlines or the maximum jitter for any task. The goal is often to schedule the task so that the undesired effects are minimized so that the controlled application is unaffected by the network.

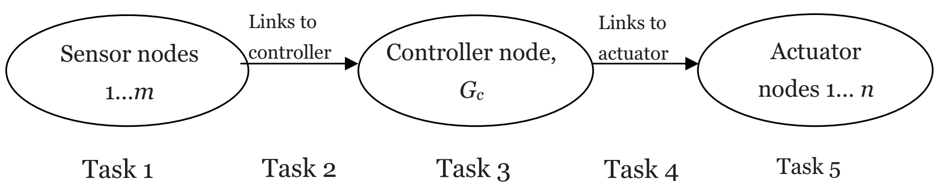

Figure 7. Typical task sequence diagram and information flows in NCS.

3.2 Network-induced effects for the control in NCS

When sensors, actuators, and controllers are distributed and interconnected by communication networks, the flexibility, reconfigurability and maintainability of the overall system are increased. However, as time delays of different types are unavoidable because of the sharing of the communication media, there are several network-induced effects that need to be considered. The overall effect of introducing the communication network to the feedback loop is that otherwise time invariant system becomes a time variant one with additional nonlinear properties. For the applications with time critical specifications, this additional time-variant behavior and nonlinearization may degrade the overall performance so that the effect has to be taken into account in the analysis and design of the system.

The most common timing effects can be categorized into control and network-induced delays, delay variation or jitter effects, transient faults, asynchronization and sampling period effects (Wittenmark, *et al.*, 1995; Sanfidson, 2000).

3.2.1 Control and network-induced delays

The control induced delay τ_c occurs while exchanging data among devices connected to the same shared medium. The control induced delay is a combination of a sensor-to-controller delay ($\tau_{sc,1},\ldots,\tau_{sc,m}$), a controller-to-actuator delay ($\tau_{ca,1},\ldots,\tau_{ca,n}$) and various computation delays (τ_{comp}). The actuator and sensor delays can be further considered as a combination of a delay due to the limited bandwidth (τ_{bw}) and a delay due to overhead in the communicating nodes (τ_{oh}). For sensor *m* and actuator *n*, the control induced delay can be summarized as:

$$\tau_c = \tau_{sc,m} + \tau_{ca,n} + \tau_{comp} \tag{1}$$

With the development in hardware of IT technology, the issue of the computational delay part of the control delay, τ_{comp}, has now become less important (in case nonrecursive control algorithm is used). The remaining effect, the communication induced delay caused by the signal transfer method adopted, is commonly known in the literature as network-induced delay τ_{nw}. The network delay between the sensor *m* and the actuator *n* can be written as:

$$\tau_{nw} = \tau_{sc,m} + \tau_{ca,n} \tag{2}$$

These different delays can be represented with a timing diagram as seen in Figure 8:

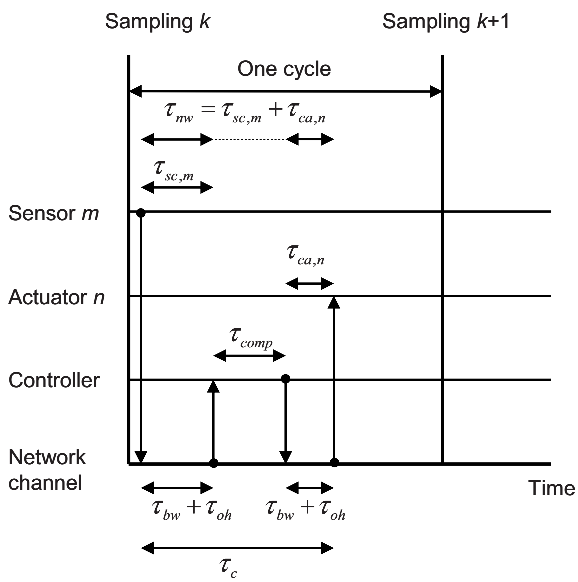

Figure 8. Timing signal in NCS.

3.2.2 Transient faults, jitter, asynchronization and sampling effects

A timing fault caused by a malfunction in a communication network with a transient duration is a transient fault. A transient fault is an unpermitted deviation due to a lost or corrupted data message transmitted in the network. As a result of a transient fault information may get lost, the control delay may increase, the state may become inconsistent or calculated output may deviate from the correct value due to the resulting vacant sampling or sample rejection. If a transient fault persists for several control periods, this may lead to a communication failure. Communication failure is a permanent interruption of a system's ability to perform a required function. If the system does not recover quickly from the failure state, sequential samples become vacant which may lead to a failure in the network. The control loop is, in fact, run in an open loop and it depends on the dynamics, the current state, control actions and upcoming disturbances, how long it is possible to avoid a system failure. Any transient fault should be avoided if possible. However, if the drift of the state during a failure is small, the loss of the control signal might not be noticed from the perspective of the process being controlled. The possibility of the occurrence of transient faults is also dependent on the network protocol as some protocols have error correction mechanisms implemented and thus no message losses can occur - the information loss is visible only though the increased network transmission delay.

Jitter can be defined as time-related, abrupt, spurious variations in the duration of any specified interval and occurs, for example, due to a clock drift, branching in the code, the scheduling algorithm or the use of specific hardware (e.g. cache memory). Jitter is often an unintentional result when scheduling a set of tasks, for instance. It is possible to define different kinds of jitter depending on the context - in scheduling there is input jitter, output jitter and queuing jitter. Jitter on a microscopic time scale (i.e. variations considerably less than the sampling period) is not a serious timing issue, whereas on a macroscopic scale it has to be taken into account and the effect becomes similar to the network-induced delay.

The incorrect choice of sampling period can also lead to undesirable effects. Conventionally, a faster sampling rate is desirable in sampled-data systems so that the discrete-time control design and performance can approximate those of the continuous system. In NCSs, however, a faster sampling rate can increase the network load which, in turn, results in a longer transmission delay of the signals and time variance. Thus, determining sampling rate that can achieve the desired system performance and tolerate the network-induced delay is important in NCS design.

One effect which occurs in the control systems is time variations due to asynchronous signals within the control loops. An example is when the communication is done over a fieldbus where the signal is sampled with a rate higher than the sampling rate of the controller and then sent to the controller node. This means that due to the sampling and synchronization of the signals to and from the I/O device to the controller, there may be a delay variation which affects the performance of the closed loop system.

Although all of these timing phenomena have to be considered when designing the NCS, the main emphasis in this work will be on the compensation and reduction of the most important effect - the network-induced delay. This effect is of particular importance in switched Ethernet type networks. In addition, all other timing issues can be seen as additional time-variant behavior and time-varying delay that affect the system.

3.3 The modeling of NCS

Research on the modeling of NCS has been carried out using continuous-time, discrete-time and sampled-data models. In addition, a more NCS-specific method called the perturbation approach has been commonly utilized.

3.3.1 Discrete time models

It is natural to study a NCS in discrete-time since in a typical NCS operation, physical signals are sampled and then transmitted on the network medium. For discrete time models, researchers often assume that the network is synchronized and that the sampling rates of the sensors, controllers and actuators are the same.

In work by Nilsson (1998), the dynamics of a remote plant are described by the following discrete state space model:

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma_0(\tau)u(k) + \Gamma_1(\tau)u(k-1) + v(k) \\ y(k) &= Cx(k) + w(k) \end{aligned} \tag{3}$$

and as a controller the following discrete controller then is utilized:

$$u(k) = -Fx(k) \tag{4}$$

where $\tau = \tau_{sc} + \tau_{ca}$ indicates the network delays at sampling time *k,* $\Phi = e^{Ah}$, $\Gamma_0(\tau) = \int_0^{h-\tau_{sc}-\tau_{ca}} e^{As} ds B$, $\Gamma_1(\tau) = \int_{h-\tau_{sc}-\tau_{ca}}^{h} e^{As} ds B$, where h represents the sampling period, A, B, and C are matrices of a continuous system. The stochastic processes $v(k)$ and $w(k)$ are uncorrelated Gaussian white noises with zero means. (It would be more precise not to move the sensor delay to the input but model the system as follows:

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma_0(\tau_{ca})u(k) + \Gamma_1(\tau_{ca})u(k-1) + v(k) \\ y(k) &= C_0(\tau_{sc})x(k) + \Gamma_1(\tau_{sc})x(k-1) + w(k) \end{aligned} \tag{5}$$

where the controller and sensor delays are not combined).

To model the network delay, a model is proposed where it is assumed that the network delay under specific network loading conditions varies according to Gaussian distribution, and different mean values in the distribution are used to model different loading cases. The transition between the network loading cases is modeled as a Markov chain. An example of a Markov chain modeling is illustrated in Figure 9 where three different network loads are represented, low (L), medium (M) and high (H), together with the arrows showing possible transitions in the load. In the figure, the delay distributions corresponding to each of the states of the Markov chain are modeled according to Gaussian distribution.

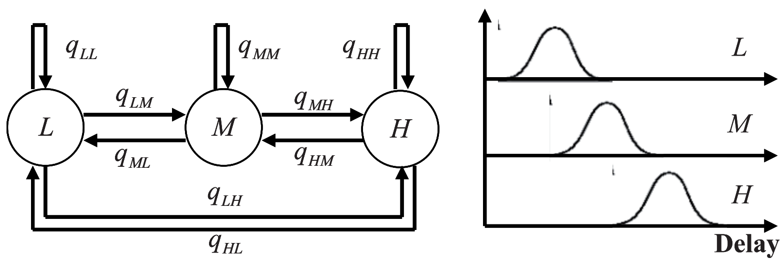

Figure 9. An example of a Markov chain modeling the state in a communication network (left), and the delay distributions corresponding to the states of the Markov chain (right). L is the state for low network load; M is the state for medium network load; and H is the state for high network load. The arrows show possible transitions in the system (Nilsson, 1998).

The model in Nilsson (1998) assumes that the network-induced delay is shorter than a sampling period. A similar approach is taken in Shousong and Qixin (2003) who use a discrete time model of a networked control system adopted from Liou and Ray (1991), but with the network-induced delay longer than a sampling period. This leads to a more complicated model as the dimension of the state space model varies. Also in Kim, *et al.* (2006), Matveev and Savkin (2004), Liu, *et al.* (2007), Zhang and Hua-Jing (2006) and Imer, *et al.* (2006) the discrete time models are utilized, and the models varies with respect to ability to take into account dropouts, whether the controller and plant are plant or event driven, etc.

3.3.2 Continuous time models

Another approach to model the NCS system is to use continuous time models. This approach is especially employed for TCP-like networks, where the flow models are used to describe the temporal network behavior. Continuous models are also used when the communication network effect is represented as an uncertainty around the nominal continuous plant and controller, (Walsh, *et al.*, 2001).

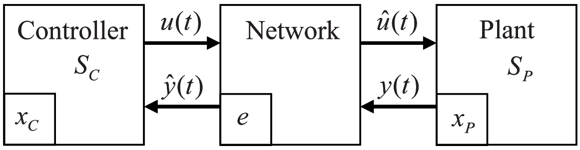

Figure 10. The NCS model with the perturbation approach, (Walsh, *et al.*, 2001).

One of the most common approaches specifically developed for NCS is to use continuous models for the plant and controller, and to represent the network effect as a block that introduces an additional error between the transmitted and received signals. Consider the NCS model in Figure 10. It consists of three main parts: the plant $S_P(A_P, B_P, C_P, 0)$ with state $x_P(t)$ and output $y(t)$; the controller $S_C(A_C, B_C, C_C, D_C)$ with state $x_C(t)$ and output $u(t)$; and the network with state $\hat{n}(t) = \left[\hat{y}(t)^T \quad \hat{u}(t)^T\right]^T$, consisting of the most recently reported versions of $y(t)$ and $u(t)$. The network-induced error is labeled as $e(t) = \hat{n}(t) - \left[y(t)^T \quad u(t)^T\right]^T$ and the combined state of the controller and plant $x(t) = \left[x_P(t)^T \quad x_C(t)^T\right]^T$. The state of the entire NCS is given by $z(t) = \left[x(t)^T \quad e(t)^T\right]^T$ and between transmission instances the dynamics of the NCS can be summarized as

$$\dot{z}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \tag{6}$$

where

$$\begin{aligned} &A_{11} = \begin{bmatrix} A_P + B_P D_C C_P & B_P C_C \\ B_C C_P & A_C \end{bmatrix}, \quad A_{12} = \begin{bmatrix} B_P D_C & B_P \\ B_C & A_C \end{bmatrix}, \quad A_{21} = -\begin{bmatrix} C_P & 0 \\ 0 & C_C \end{bmatrix} A_{11}, \\ &A_{22} = -\begin{bmatrix} C_P & 0 \\ 0 & C_C \end{bmatrix} A_{12} \end{aligned} \tag{7}$$

Without the network $e(t) = 0$, and hence the dynamics reduce to $\dot{x}(t) = A_{11} x(t)$. It is assumed that the controller has been designed ignoring the network, and therefore the eigenvalue of A_{11} has negative real parts.

The temporal variation of network delay is presented as local variation of signals. The actual network is not directly modeled, but its influence is visible indirectly in the process and controller states. The network could be modeled, for example, with colored noise by inserting some dynamics into A_{22}, for example $A_{22}+E$.

3.3.3 Sampled-data models

A recent NCS modeling approach is to use sampled-data models, which is the most accurate representation of an actual situation. In this kind of setup, the model is divided into two subsystems: the first represents a distributed continuous plant to be controlled with sensors and actuators, while the second represents a piecewise constant digital controller. The network channel is modeled as a delay system with time variant properties. In Hu, *et al.* (2007), the NCS is modeled as a hybrid system, which involves a continuous plant and event-driven or time-driven devices. The delay free plant is described as

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{8}$$

and a digital state-feedback controller for delay free plant is:

$$\tilde{u}\left[t^k\right] = Fx(t^{k-1}) \tag{9}$$

where *A*, *B*, *F* are real matrices with appropriate dimensions and $\left[t^k, k = 1,2,\ldots,\right]$ are the sampling instants satisfying $t^{k+1} - t^k = h$, where h is sampling period. If every control action $\tilde{u}$ is held by a zero-order holder and only valid over the interval $\left[t^k + \tau_{sc}^k + \tau_{ca}^k, t^{k+1} + \tau_{sc}^{k+1} + \tau_{ca}^{k+1}\right]$, then the event driven controller can be described as

$$u(t) = \tilde{u}\left[t^k + \tau_{ca}^k + \tau_{sc}^k\right] \tag{10}$$

The superscript k in τ_{ca}^k and τ_{sc}^k is used to describe time-varying nature of the delay. Under the condition of $h + \tau_{sc}^{k+1} + \tau_{ca}^{k+1} \le 2\left(\tau_{sc}^k + \tau_{ca}^k\right)$ the effect on the plant with an event-driven actuator and time-driven sensor can be written as

$$\dot{x}(t) = Ax(t) + Bu(t - \tau_{sc}^{k+1} - \tau_{ca}^{k+1}) \tag{11}$$

The NCSs with packet dropout can be described as the case where the control action is not updated during the time interval of packet dropout increasing the delay to be longer than the sampling period. Similar models for the NCS were used by several other authors, see Yi, *et al*. (2007), and Yue, *et al*. (2005).

3.4 Control strategies for NCS

In the controller design for NCS, the objective is to obtain a control law that is able to compensate for the effect of the network so that the application exhibits an acceptable performance, even under worst-case fluctuations of QoS.

The issue of the effect of the delay on the control performance is itself not new; time delay (or dead time) arising within the plant itself is a feature of many feedback control systems. This type of delay can be called natural delay or plant delay and there has been considerable research on the control of systems with plant delays.

In a NCS, however, there is time delay caused by the signal transfer method adopted. Because of their discrete and distributed nature, network-induced delays are different from the plant delays and computational delays that have been studied in the past. Often, the conventional control theories with many ideal assumptions, such as synchronized control and non-delayed sensing and actuation, must be re-evaluated before they can be applied to NCSs.

In some circumstances, it is possible to modify the network performance so that the effect of the network transmission delays for the controller becomes similar to the traditional delays. In these cases, the problem of dealing with network-induced delays can be simplified to a similar problem of dealing with traditional delays and it becomes possible to use well known tools for linear sampled-data systems for the analysis and design of NCSs. Indeed, some studies of time-delayed-system analysis and design although not specific to NCSs, such as Michels, *et al*. (2005), Zhang, *et al*. (2004), Jing, *et al*. (2004), Chen, *et al*. (2004) and Richard (2003), provide results that are applicable to NCSs.

Several compensation strategies have been proposed based on the information available from the network or on assumptions about the application. The proposed control strategies vary from the traditional pole placement techniques, PID control, gain scheduling and the state feedback controls to more sophisticated control schemes such as MPC and H_∞ control.

3.4.1 PID control

Chow and Tipsuwan (2003) proposed an approach for networked DC motor control systems using a controller gain adaptation in the PI controller to compensate for the changes in QoS requirements. With the numerical and experimental simulations and prototyping, the authors demonstrate the feasibility of the proposed adaptation scheme to handle a network QoS variation in a control loop. Another study on PID control in NCS was carried out by Eriksson (2005), where PID tuning rules for networked control systems were proposed. In Tian and Levy (2008), PD type controllers are used to compensate for packet dropouts using knowledge on past control signals and the predictive property of the derivative term.

3.4.2 Smith predictor type approaches

In Chen, *et al.* (2007), a Smith predictor is proposed to compensate for the round trip delay of networked control systems (NCSs). An analytical dynamic TCP model created using fluid-flow and stochastic differential equations was adopted. The authors also show that the stability of the closed-loop system is guaranteed up to some maximum upper bound of the round trip delay. The neural net-based Smith predictor approach that utilizes the similar TCP model is also presented in Lin and Chen (2006). The Smith Predictor is also applied in Bauer, *et al.* (2001), where authors show that long access delays are not necessarily detrimental to the stability of the local area network (LAN) in embedded control systems.

3.4.3 Standard LQG control

One of the earliest works where a control approach was proposed specifically for NCS was by Nilsson (1998). Nilsson considered the problem of controlling a continuous-time linear plant sampled at a constant rate over a network with sensor-to-controller and controller-to-actuator delays. With an assumption that the actuator is event-driven, the LQG-optimal controller was derived for a network where the time delays are independent from sample to sample. An additional important result of Nilsson's work was that the separation principle applies and that the optimal controller is the combination of a state feedback controller and a Kalman filter. One limitation in his work is that the network-induced delay is assumed to be smaller than the sampling period. In Shousong and Qixin (2003), stochastic optimal controllers of the networked control systems whose network-induced delay is longer than a sampling period were designed for the two cases of the system - either full state information or partial state information. The separation theorem is proved to still hold in such networked control systems.

Hristu-Varsakelis and Zhang (2008) use an LQG design method for NCSs, which are subject to medium access constraints and known delays. The approach forgoes the use of ZOH elements in the loop; instead, the controller and plant 'ignore' sensors and actuators which are not granted medium access. The selection of communication sequences is decoupled from the choice of controller by relaxing the requirement for jointly optimal control and communication, thereby simplifying the identification of useful communication patterns. Authors also showed that it is always possible to design periodic communication sequences that preserve the detectability and stabilisability of the LTI plant in the presence of communication constraints.

3.4.4 Optimal control based approaches

Optimal control based approaches have also been successfully applied. In Imer, *et al.* (2006), two different types of networking protocols are considered - the TCP and the UDP structure. These protocols differ with respect to the availability of acknowledgment (ACK) signals; under a TCP structure, the ACK signals exist, unlike the UDP structure where no ACK packets are present. A methodology is presented to stabilize the system while minimizing a quadratic performance criterion when the information flow between the controller and the plant is disrupted due to either link failures or packet losses. No transmission delay is considered. For the plant described by

$$x_{k+1} = Ax_k + \alpha_K Bu_k + w_k \qquad k = 0,1,\ldots, \tag{12}$$

the proposed quadratic function has the following form:

$$J_{\{\mu_0(I_0),\mu_1(I_1),\ldots,\mu_{N-1}(I_{N-1})\}} = E\left\{x_N^T F x_N + \sum_{k=0}^{N-1}(x_k^T Q x_k + \alpha_k u_k^T R u_k) \middle| \{\mu_0(I_0),\mu_1(I_1),\ldots,\mu_{N-1}(I_{N-1})\}\right\} \tag{13}$$

where μ_i is an admissible control policy under information vector I_i, N is decision horizon, and α_K is a stochastic parameter that is used to model the losses in the communication and is assumed to follow either Markov or Bernoulli process depending whether it is TCP or UDP. The stochastic process $\{\alpha_K\}$ models the unreliable nature of the link from the controller to the actuator. Basically, in the UDP case, α_K is considered zero when this link fails, i.e., the control packet is lost, and $\alpha_K = 1$ otherwise. Principally, this corresponds to the zero control action by the actuator in case of packet loss (ZOH is a more obvious choice). For the TCP case, the stochastic parameter α_K follows the independent and identically distributed Bernoulli process with $P[\alpha_K = 0] = \alpha$, and $P[\alpha_K = 1] = 1 - \alpha := \overline{\alpha}$. In addition, sufficient conditions for the existence of stabilizing optimal controllers are derived.

Also in Shousong and Qixin (2003), stochastic optimal controllers of networked control systems whose network-induced delay is longer than a sampling period are designed for the system with either full state information or partial state information. For both cases controllers are derived that minimize the quadratic cost function. In addition, the optimal estimator of the system state is presented when the system has partial state information and the network-induced delay is longer than a sampling period. The separation theorem is proved to still hold in such networked control systems. Also in Lian, *et al.* (2003), the optimal control theory is used to obtain control law for NCS. In this work, the authors specifically address the MIMO type of NCS and considered multiple distributed communication delays. The time delays between sensor-controller and controller-actuator and the time skews at different devices' sampling instants were considered and included in the discrete-time MIMO model. Based on the time-delay modeling algorithm and considering the case of constant communication delays, a delayed state-variable model was formulated for the standard LQR (Linear-Quadratic-Regulator) optimal controller design. The proposed control algorithm utilized the information of delayed signals and improves the control performance of a control system encountering distributed communication delays.

3.4.5 Model predictive control

The model predictive control approach has been proved effective for the control of the NCS. The benefits of the MPC algorithm respond well to timing difficulties such as network-induced delay and missing values. On the other hand clock synchronization is often required. In addition, the MPC approaches require solving the optimization problem at each sampling instant which significantly increases the computation delay.

Tang and de Silva (2006) extended the Generalized Predictive Control (GPC) algorithm to compensate for data-transmission delays. The developed networked control strategy consists of a plant and the sensor and actuator nodes which are located at a remote site and connected through a network to a GPC module located at a local monitoring and control station. The sensor node, the actuator node, and the controller have identical sampling periods and the three nodes are synchronized. An estimator is used to estimate the delayed or missing sensor data. Once the delayed or missing system responses are estimated, the next step of the developed networked control strategy is to predict the required *k*-step-ahead future control signals that will drive the system to track a desired trajectory. As a distinction from a typical receding-horizon GPC strategy, where only the first input is implemented, the presented networked control strategy uses all of the elements to compensate for delayed or missing control signals in the controller-to-actuator communication lines. A model-predictive-control strategy with a timeout scheme was also developed by Kim, *et al.* (2006) and tested on an open-loop unstable magnetic-levitation (maglev) test bed. In Huang, *et al.* (2009), a two level hierarchy is employed in the design of NCS with a bounded random transmission delay. At the lower level, a local controller is designed to stabilize the plant while at the higher level a remote controller with modified dynamic matrix control (DMC) algorithm is implemented to regulate the desirable set-point for the local controller. Under the assumption that the closed-loop local system is described by one Finite Impulse Response (FIR) model of an FIR model family, the robust stability problem of the remote DMC controller is investigated. For other results on the MPC implementation for NCS see Srinivasagupta, *et al.* (2004), Liu, *et al.* (2006) and Li, *et al.* (2009).

3.4.6 Robust control based on *H2/H∞* norm minimization

Recently, norm minimization based approaches have been applied to the delay compensation of the NCS. In Yue, *et al.* (2005), the authors consider the design of robust H_∞ controllers for uncertain networked control systems. The same NCS hybrid model, as presented by the Equations (10) and (11), is used to describe the performance of the NCS. The H_∞ minimization problem is constructed and solved as an LMI optimization problem, obtaining a state feedback. The information of the lower bound of the network-induced delay is utilized in the formulations. In Georgiev and Tilbury (2006), the control problem is solved using H_2 optimization techniques. As a network model discrete time model is used, and the data is assumed to be transmitted in a multipoint packet structure where several data points from the same sensor or actuator are combined together. The control problem for the multipoint-packet system is shown to be equivalent to a multirate control problem which, in turn, is reduced to a synthesis problem with a constraint on the feedthrough matrix D. It is possible to reduce the computation time as well as the network traffic with the structure. The control law obtained via H_∞-norm minimization has also been obtained in Gao, *et al.* (2008), who use a continuous state space model with two consecutive delays in the state, which are used to describe different segments of the network. Li, *et al.* (2010) have studied the observer-based H_∞ control for nonlinear NCS with random packet losses. A nonlinear sampled data model is used to model the NCS, and a Bernoulli distributed white sequence with a known conditional probability distribution is used to model the random packet loss. Other work on H_∞ control can be found in Peng and Tang (2009), Gao and Chen (2008) and Ishii (2008).

3.5 Stability and performance analysis in NCS

As a part of control design, the stability and performance analysis of the obtained NCS system should be performed. The stability analysis of time delay systems, especially for time variant delay systems as is the case for NCS, is considerably more complicated than that of a delay-free system. One of the main reasons is that when a delay appears in the system, the characteristic equation becomes a transcendental equation containing polynomial and exponential functions instead of pure polynomial ones. Even a transcendental equation cannot be defined for time-varying systems.

By stability, it is traditionally meant that for any bounded input over any amount of time, the output will also be bounded (Bounded Input Bounded Output (BIBO) stability). This means that for a delay free linear system to be stable, all of the poles of its transfer function must satisfy some criteria depending on whether a continuous or discrete time analysis is used. In continuous time, a system is stable if the poles of the transfer function lie strictly in the closed left half of the complex plane. In discrete time, the Z-transform is used and a system is stable if the poles of this transfer function lie strictly inside the unit circle. For a delayed or time variant system, the concept of poles is not easily defined (Kamen, 1988) and thus other approaches should be used. In this section, the fundamental results on stability in frequency and time domains are summarized.

3.5.1 Frequency domain approaches

In cases when the delay variation is not significant and it is bounded, a frequency domain approach may provide guidelines on the stability of the overall NCS system. Most of the frequency domain stability results are based on the application of the small gain theorem and its modifications. Although the stability result obtained from the small gain theorem is often conservative as the phase of the system is not considered, the stability criterion is simple and can be easily checked from the closed loop Bode margin or Nyquist plots. Gu, *et al.* (2003) used a small-gain approach for proving the stability of systems with time-varying delays, where the time variation of the delay parameters was bounded and its variation speed was strictly less than 1, or $|\dot{\tau}(t)| < 1$. Kao and Lincoln (2004) extended the result to include arbitrary fast delay variations within the upper bounded delay value. The result was expressed as the following theorem: Consider linear time invariant system with uncertain time-varying time delay in the feedback as presented in Figure 11. Assuming that the closed loop system with continuous-time, time-invariant plant $P(s)$ and controller $C(s)$ is stable for zero delay, then the system is stable for any time-varying but bounded delay defined by:

$$\Delta(v)(t) = v(t - \tau(t)) \qquad 0 \le \tau(t) \le \tau_{\max}$$

If

$$\frac{P(j\omega)C(j\omega)}{1 + P(j\omega)C(j\omega)} < \frac{1}{\tau_{\max} \cdot \omega}, \forall \omega \in [0, \infty] \tag{14}$$

where Δ is the delay operator that is used to delay the incoming signal v by $\tau(t)$. In deriving the stability criteria, the idea is to calculate an upper bound of the induced gain of $\Delta(v)$, which then allows the application of an input-output analysis such as the small gain theorem.

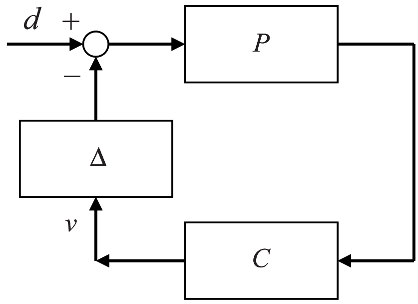

Figure 11. Linear time invariant system with uncertain time-varying time delay in the feedback loop (Kao and Lincoln, 2004).

For the discrete time plant $P(z)$ and $C(z)$ the system is stable for any time-varying delays defined by:

$$\Delta(v)(n) = v(n - \delta(n)) \qquad \delta(n) \in \{0,1,\ldots,N\} \tag{15}$$

if

$$\frac{P(e^{j\omega})C(e^{j\omega})}{1 + P(e^{j\omega})C(e^{j\omega})} < \frac{1}{N \cdot \left| e^{j\omega} - 1 \right|} \qquad \forall \omega \in [0, \infty] \tag{16}$$

3.5.2 Time-domain approaches

Often, however, the frequency domain techniques may not prove to be applicable, and the stability results adopted from nonlinear and time variant control theory are used. The stability results specifically addressed towards NCS utilize Lyapunov stability theory-based approaches, such as the Lyapunov second method (direct method), the Lyapunov-Razumihin approach, and the Lyapunov-Krasovskii approach. For a detailed presentation see the survey by Richard (2003).

The stability can be studied based on the standard Lyapunov second method, if the explicit model of NCS is used, i.e. the delay is not visible directly from the plant/controller equations as presented in Section 3.3.2 or in Montestruque and Antsaklis (2004). The Lyapunov second method is used to study the stability of the following nonlinear system:

$$\dot{x}(t) = f(x(t)), \; f(x_e) = 0 \tag{17}$$

where x_e is some equilibrium point. The idea behind the Lyapunov second method is to find a function $V(x)$, (called Lyapunov function) so that that $V(x)$ is positive definite, $V(x) > 0$, $x \neq x_e$, $V(x_e) = 0$ and the derivative of $V(x)$ along the trajectories of (17) is negative definite:

$$\dot{V}(x) = \frac{\partial V^T}{dx}(x) f(x) < 0, \qquad x \neq x_e \tag{18}$$

Then $f(x)$ is stable around the equilibrium point and every trajectory of $\dot{x}(t) = f(x)$ converges to x_e as $t \to \infty$. Graphically, the method can be interpreted as follows: Equation (18) corresponds to an inner product $\left\langle \frac{\partial V}{\partial x}(x), f(x) \right\rangle$, and if its value is negative, then the cosine of the angle between $\frac{\partial V}{\partial x}(x)$ and $f(x)$ is negative, or the angle is greater than 90 degrees, and the values $x(t)$ converge toward x_e. Graphically, this principle behind the second Lyapunov method is presented in Figure 12.

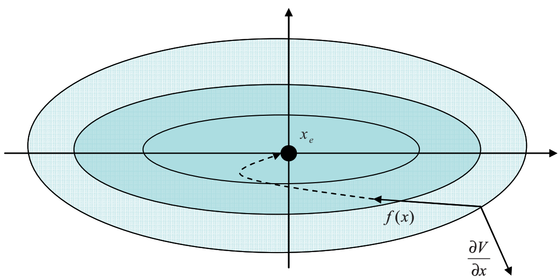

Figure 12. Principle of the Lyapunov second method.

The NCS research where the Lyapunov second method has been used to study the stability can be found in Montestruque and Antsaklis (2007) and Montestruque and Antsaklis (2004).

The Lyapunov-Razumihin, and the Lyapunov-Krasovskii methods are generalizations of the Lyapunov direct method. The Lyapunov–Razumikhin approach allows to prove stability of systems with bounded but freely fast time-varying delays. In the approach functions with certain properties are sought and if the function is found, the respective system is considered to be stable. The Lyapunov-Krasovskii approach allows to prove stability of time-delay systems where the delay parameters are bounded both in length and time variation. Instead of a function, the functionals with specific properies are sought (Niculescu, 2001). The Lyapunov-Krasovskii approach generally leads to a less conservative result, however, as it requires the manipulation of the functionals, it is more difficult to apply.

The system that is studied in both cases includes a delay and is represented as retarded functional differential equations (RFDEs) given as

$$\begin{aligned} &\dot{x}(t) = f(t, x(t+\theta)), && t \geq t_0, \theta \in [-\tau_{\max}, 0] \\ &x(t_0 + \theta) = \phi(\theta), && \theta \in [-\tau_{\max}, 0] \end{aligned} \tag{19}$$

where $x(t)$ is the state variable of the system, θ is the delay and $\tau_{\max}$ is the maximum value of the delay of the system.

In the Lyapunov-Krasovskii method, a continuously differentiable functional V is sought so that

$$\begin{aligned} &u(\|\phi(0)\|) \leq V(t, \phi) \leq v(\|\phi(0)\|_c), \\ &\dot{V}(t, \phi) \leq -w(\|\phi(0)\|) \end{aligned} \tag{20}$$

where u, v and w are continuous nondecreasing functions, $u(p)$ and $v(p)$ are positive for $p > 0$, and $u(0) = v(0) = 0$. In the above equation, ϕ is the continuous function $\phi : [-\tau_{\max}, 0] \to \mathbb{R}$, and $\|\phi\|_c$ is the continuous norm of ϕ:

$$\|\phi\|_c = \max_{-\tau_{\max} \leq \theta \leq 0} |\phi(\theta)| \tag{21}$$

If the functional V exists, then the trivial solution of (19) is uniformly stable. Therefore, the functional V measures the deviation of the system from the trivial solution.

In the Lyapunov-Razumikhin method, the functions are studied instead of functionals. The main idea behind the Lyapunov-Razumihin is to focus on a function that is representative of the size of the solution $x(t)$ and function $V(x)$. For such function $V(x)$, the following function serves to measure the size of x_t:

$$\overline{V}(x_t) = \max_{\theta \in [-\tau_{\max}, 0]} V(x(t+\theta)) \tag{22}$$

If $V(x(t)) < \overline{V}(x_t)$, then $\dot{V}(x) > 0$ does not make $\overline{V}(x_t)$ grow. For $\overline{V}(x_t)$ not to grow, it is only necessary that $\dot{V}(x(t))$ is not positive whenever $V(x(t) = \overline{V}(x_t)$.

More formally, if there exists a continuous differentiable function $V(t,x)$ so that

$$u(\|x\|) \leq V(t,x) \leq v(\|x\|), t \in \mathbb{R}, x \in \mathbb{R}^n \tag{23}$$

and the derivative of V along the solution $x(t)$ of (19) satisfies

$$\dot{V}(t, x(t)) \leq -w(\|x(t)\|) \tag{24}$$

whenever $V(t+\theta, x(t+\theta)) \leq V(t, x(t)), \theta \in [-\tau_{\max}, 0]$ then the system (19) is uniformly stable.

In Fridman, *et al.* (2004), the Lyapunov-Krasovskii approach has been applied to ordinary sampled-data systems, which have been modeled as linear continuous time systems with time-varying input delays under the constraint $|\dot{\tau}(t)| \leq 1$. Also in Yu, *et al.* (2005), the formulation of stability conditions for a NCS modeled as a sampled-data system with continuous-time plant and a piecewice constant controller was based on the Lyapunov-Krasovskii technique with an extension that the technique also aplies for nonlinear systems and no iterative procedure to determine the tuning parameter is involved. In Tang, *et al.* (2008), a delay dependent Lyapunov-Krasovskii type functional is proposed to study the stability the state space system with a memoryless state feedback controller. In Cao, *et al.* (2008), a delay dependent stability of MIMO systems is studied based on the Lyapunov-Krasovskii techniques. In Huang and Nguang (2008), the stability of a linear uncertain networked control system with random communication time delays is studied with the Lyapunov-Razumikhin method. In Jiang and Han (2008), the robust stability of uncertain linear systems with an interval time-varying delay is studied with a Lyapunov–

Krasovskii functional. Other results on stability can be found in Zhang, *et al*. (2005), Cheng and Savkin (2007) and Walsh, *et al*. (2002b).

3.6 Control of network methods for a NCS

Control of network problem in a NCS was formulated in Chapter 1 as follows: given a set of processes to be controlled and a communication network with limited resources, design and schedule the network tasks so that the overall control performance is optimized. Control of network problem in a NCS, therefore, is often reduced to either network real-time scheduling or bandwidth minimization.

3.6.1 Real-time scheduling

One possible methodology to control the overall performance of a NCS is to find a maximum allowable dead bound (MADB) for sampling, for which stability can be guaranteed, and then to find an appropriate network scheduling method that limits the network-induced delay to less than the MADB. Much research has been carried out around this matter. For a review on determining a maximum allowable delay bound for scheduling see research done by Kim, *et al*. (2003). The latest result on MADB can be found in Ma, *et al*. (2007), where the authors obtain the MADB guaranteeing the stability of the NCS for the systems with multi-step delay. Scheduling methods that utilize MADB have also been proposed by several authors such as Park, *et al*. (2002) and Walsh, *et al*. (2002a).

For existing network protocols in Juanole and Mouney (2006), the authors analyze the effect of different message scheduling on the stability of the process and control performance. The message scheduling policies of three local area networks are considered: CAN, FIP and ARINC. A simulation study is provided where the effect of an adopted message transfer policy (with loss control or without data loss control) and the task scheduling algorithm (Rate monotonic (RM) algorithm for independent tasks and RM for tasks sharing resources) is studied. In Lee, *et al*. (2005), authors focus on a quality-of-service-based (QoS) remote control scheme for networked control systems via the Profibus token passing protocol. To ensure the control performance of a NCS, the network delay is maintained below the allowable delay level. The network delay is affected by protocol parameters, such as target rotation time, and an algorithm for the selection of the target rotation time using a genetic algorithm to ensure that the QoS of the control information is presented.

3.6.2 Bandwidth minimization

Control and minimization of the bandwidth is studied in Yook, *et al*. (2002). The authors propose a framework for distributed control systems in which the estimators are used at each node to estimate the values of the outputs at the other nodes. The estimated values are then used to compute the control algorithms at each node. When the estimated value deviates from the true value by more than a pre-specified tolerance, the actual value is broadcast to the rest of the system. The minimization of the traffic was also studied by Otanez, *et al*. (2002), where they reduced the network traffic with an adjustable deadband strategy by suppressing the number of updates sent to the network when the absolute difference between the latest sent signal and the current measured signal is below a certain deadband value.

3.7 Control strategies specifically addresses for switched Ethernet based networks

Only few studies are found where control strategies have been developed particularly for the switched Ethernet based networks. Some exceptions exist such as research done in Lee, *et al*. (2006) and Tang and de Silva (2006).

In Lee, *et al*. (2006), the authors present an analytical performance evaluation of the switched Ethernet with multiple levels from timing diagram analysis, and the estimation of the worst case delay of a real-time industrial Switched Ethernet with multiple levels. In addition, the authors provide an experimental evaluation from an experimental test bed of networked control system. In their experimental evaluation, the authors show the comparison of process responses with the PID controller via a direct connection, traditional Ethernet, and switched Ethernet. This result shows that the performance of the switched Ethernet with multiple levels may be acceptable industrial network with real-time requirements. Tang and de Silva (2006) present the compensation for transmission delays for a switched Ethernet-based network using variable horizon predictive control.

3.8 Summary of existing control methods for NCS

Control methods for NCS methods presented in Sections 3.4 and 3.5.2 differ with respect to how information on network performance has been obtained; how the control method utilized is suitable to handle time variant behavior caused by the network; and whether or not other disturbances/uncertainties related to the process or process measurement have been considered. In this section, the methods presented in the review are classified with respect to these criteria. After classification, those methods that are missing will be addressed in subsequent chapters of this thesis.

For the first criterion, how information on the network has been obtained, three cases will be studied. Network transmission delay is considered as a parameter that describes network performance level. In the first case, the network delay is measured directly. If network performance can be measured precisely, and the measurement from the controlled application is accurate, the requirement for the control method are different compared to the case when only a rough estimate about network performance is available, and significant measurement errors occur. It is relatively straightforward to measure a round trip transmission delay as more network-related instruments or messaging is required to measure end-to-end delay either from the process to controller or from the actuator to the controller. The second case describes whether the network transmission delay has been estimated using a model, and the estimate provided by the model is used in the control. The third case describes research where there has been no emphasis on how to obtain information on network delay - this information is considered known by assumption.

For the second criterion, how the control method utilized is suitable to handle time variant behavior caused by the network, two cases will be studied. In the first case, the authors address time variant behavior explicitly. As a high time variance poses significant challenges for the control, special emphasis has been paid on how fast the delay varies, and methods from the time variant control theory are applied. In the second case, the proposed method does not address time variance explicitly. Even though a NCS is by definition a time variant system, the use of the method where time variance is not considered explicitly is also often sufficient when delay variations are low compared to the value of the delay.

For the third criterion, how process related uncertainties have been considered, two cases will be studied. In the first case, the process disturbances, model uncertainties measurement errors, measurement noise or other non-network related issues are included in the study. Even though it is important to concentrate on network performance, in many cases challenges induced by the network cannot be distinguished from those coming from the process and therefore both should be considered together. In the second type of research, uncertainties coming from the process have not been discussed separately.

The methods presented in Sections 3.4 and 3.5.2 are classified with respect to the above mentioned criteria in Table 1 and Table 2. Table 1 presents a classification with respect to the source of information on delay and Table 2 shows the classification with respect to the extent of delay variation and uncertainties related to process or instrumentation. In Table 1, UBD stands for upper bound delay; LBD for lower bound delay; τ is the network transmission delay; τ_{tv} is time-varying network transmission delay; and $\Delta\tau$ is variation of network transmission delay.

It can be seen from Table 1 that the delay measurement often relies on the round trip time measurement due to its easy implementation. However, for the NCS the RTT may not be adequate and thus instrumentation for measuring the end-to-end delay should be developed.

Table 1. Classification of control methods for a NCS with respect to source of information on delay.

Study	**Measured τ**	**Network model for τ**	**Assumption on τ**
Classical control			Constant
Chow and Tipsuwan (2003)		QoS model as a function of UBD	Value of τ, distribution uniform
Eriksson (2005)			UBD and τ known
Chen, *et al.* (2007)		Dynamic TCP model as a function of loading factor	
Lin and Chen (2006)		Dynamic TCP model as a function of loading factor	τ follows uniform distribution
Bauer, *et al.* (2001)			τ known
Nilsson (1998)			τ distribution known
Shousong and Qixin (2003)			τ distribution known (uniform)
Hristu-Varsakelis and Zhang (2008)			τ known
Imer, *et al.* (2006)			Link failure probabilities known
Lian, *et al.* (2003)			$\Delta\tau$ known
Tang and de Silva (2006)	RTT		
Kim, *et al.* (2006)	RTT		UBD known
Huang, *et al.* (2009)			UBD known
Srinivasagupta, *et al.* (2004)	RTT		
Liu, *et al.* (2006)			UBD known
Yue and Han (2005)			UBD and mean delay known
Gao, *et al.* (2008)			UBD known, $\dot{\tau}$ known
Gao and Chen (2008)			UBD and LBD known, distribution random
Ishii (2008)			Package loss rate known

It can also be concluded from the table that no research is available where a network model is used to estimate network-related properties, with some notable exceptions: a dynamic TCP model for TCP/IP communication and a more general QoS model. Even though they may describe the network around some operation point (network traffic conditions), more generally, they will probably not give an adequate overall picture. A dynamic model of a TCP is based on a stochastic differential equation, and its step response with respect to the network traffic increase around some operating point closely corresponds to a step response of oscillative dynamic process. As a network is inherently a discrete system with possible abrupt changes in delay, a smooth dynamic response coming from the differential equation is not often present. While the QoS model gives an indication on the performance of the network, in order to calculate it an upper bound delay should be known.

As can be seen from Table 2, research where a time invariant Smith predictor has been used to compensate for the network transmission delay effect has been classified as a slowly time varying system since it is known that that the performance of Smith predictor cannot be guaranteed for fast time variant systems, particularly if the magnitude of the delay variation and the frequency are high. In some cases, the variation of time delay has not been presented in connection with simulations; however, the extent of variations can be determined from the step responses. (In the case of a fast variable delay, the step response will also experience fluctuation during rise.)

In Table 2, Yue and Han (2005) and Gao, *et al.* (2008) are classified as a time variant since results coming from the time-variant control theory, such as the Lyapunov–Krasovskii theorem, are applied. In Tang and de Silva (2006) and Kim, *et al.* (2006), the authors verify their approach in an experimental platform, which clearly exhibits time varying behavior, and thus it also was classified as time varying.

In the process related uncertainty column of Table 2, it can be seen that some of the research considers uncertainties related to the process; in Huang, *et al.* (2009), for example, several uncertainty types are lumped - both coming from the network and model; however, more progress in that area is needed particularly in incorporating network related information and control theory more closely.

It is worth noticing that also other criteria can be also found such as network type, control method, occurrence of packet losses and delay distributions, etc. However, those presented are the smallest set that categorizes control methods for a NCS with respect to the available network information, process uncertainties and use of the control method in a time variant environment.

Table 2. Classification of control methods for a NCS with respect to the extent of delay variation and consideration of process related uncertainties.

Study	**Uncertainties on y considered**	**Time varying delay, τ_{tv}**	**τ slowly time varying or constant**
Classical control			Constant
Chow and Tipsuwan (2003)			Slowly τ_{tv} *
Eriksson (2005)			Slowly τ_{tv} *
Chen, *et al.* (2007)			Slowly τ_{tv} *, **
Lin and Chen (2006)			Slowly τ_{tv} **
Bauer, *et al.* (2001)	Model uncertainty		Slowly τ_{tv} **
Nilsson (1998)			Slowly τ_{tv} transition according to Markov process
Shousong and Qixin (2003)			Slowly τ_{tv} *
Hristu-Varsakelis and Zhang (2008)	Gaussian disturbance		Slowly τ_{tv}, $\Delta\tau = 3T$
Imer, *et al.* (2006)	zero-mean random disturbance with known variance		Slowly τ_{tv} *
Lian, *et al.* (2003)	Noise		τ_{tv} (simulations with constant delays)
Tang and de Silva (2006)		τ_{tv}	
Kim, *et al.* (2006)	Gaussian noise	τ_{tv}	
Huang, *et al.* (2009)	Model uncertainty (gain and dead time)		Slowly τ_{tv} *
Srinivasagupta, *et al.* (2004)			Slowly τ_{tv} *
Liu, *et al.* (2006)		τ_{tv}	
Yue and Han (2005)		τ_{tv}, $\dot{\tau}$ restricted	
Gao, *et al.* (2008)		τ_{tv}, $\dot{\tau}$ restricted	
Gao and Chen (2008)		τ_{tv}	
Ishii (2008)	Periodic disturbance	τ_{tv}	

*The magnitude of variation determined from the step response graphs

**A time invariant Smith predictor is used

4 Advanced control methods for a NCS

Based on the summary presented in the previous chapter, it can be concluded that more progress is needed toward incorporating network instrumentation and control approaches more closely, as well as network-related model and control approaches. Additionally, time-variant behavior should be studied more closely, concentrating the challenges imposed by time-variance. A fourth issue that needs more investigation is incorporating network-based models and uncertainties related to the process. All four of these issues will be addressed in details in subsequent chapters; however, a short introduction on these is presented here.

4.1 Control of NCS using network end-to-end delay measurements

In Chapter 5, a Smith predictor based approach will be presented for compensating measured end-to-end transmission delay to combining the network instrumentation information in control more closely.

The delay measurement will be obtained from the timestamps of network packages containing the control and measurement information transmitted over a switched Ethernet network. In order to obtain a valid network delay value, all devices connected to the network (controllers, actuators, and sensors) are synchronized using the IEEE 1588 protocol.

The idea behind the presented methodology for obtaining a network delay is straightforward; also, a utilized control method is well known for delay compensation, the main novelty is in integrating them and testing the approach with the simulations as well as on a real experimental platform.

The main benefit of the approach is in its simplicity, in the case when the devices connected to the network support IEEE 1588 synchronization, the main effect induced by introducing the network in the feedback loop is handled, and control design can be made using standard control theory tools. The main problem is related to its implementation since relatively few vendors support IEEE 1588 synchronization. From the control theoretical point of view, there are known limitations of the Smith predictor in case of a model mismatch and measurement noise sensitivity.

4.2 Time varying control method for a NCS

It will be assumed that the network transmission delay measurements are available, and the time variant observer and control design with the polynomial systems theory will be proposed.

The polynomial systems theory is well established for time-invariant differential and difference linear systems. Also, a time variant case can be handled in a relatively straightforward manner with the same algebraic machinery by considering the noncommutativity of time variant systems.

The main idea behind the theory is that it utilizes the algebraic properties of polynomials with real or complex coefficients. The theory is computational in nature, and it uses the fact that the ring of polynomials over a field in an operator normally satisfies a division algorithm, which can be used to find common factors and to manipulate polynomial matrices into suitable canonical forms in an algorithmic way. Compared to the classical control theory where transfer function techniques based on Laplace- or Z-transforms is used, the theory has the advantage that it inherently handles the cancellation problem of common poles and zeros, which is one of the main drawbacks of standard techniques. Compared to the state space approach, the main drawback of which is in the lack of transparency and the lack of apparent relationship to the frequency response methods, the polynomial systems based approach retains many common features with the classical transfer function techniques. Also, the mathematical machinery needed in the state space approach is rather heavy and the treatment of interacting systems is strikingly awkward. The theory is principally the same behavior theory developed by Willems (1986), but with the differentiation in terminology where relations are considered as behaviors.

The most important drawback is that essential operations must be carried out in a rather weak algebraic structure in the ring of polynomials since equations in polynomials can be generally not be solved with respect to an unknown signal.

Another drawback is that there is a lack of established software tools, with some exceptions.

In this work, there is no contribution to the theory but to the application of polynomial theory to the control of a time varying NCS.

4.3 Control of a NCS using delay estimates obtained from a network model

The estimate for the delay used in the method for control of a NCS is obtained using a model network calculus theory. The obtained model is used in calculating the upper bound of the network delay which, in turn, is used for control synthesis.

The modeling consists of two phases: in the first phase, a methodology for delay computation over a single switch is presented by defining the upper bound delays over its basic components (multiplexers, demultiplexers, and queues); in the second phase, the upper bound delay over a switched Ethernet network consisting of several switches is calculated.

The obtained upper bound delay value is used in the controller synthesis as a parameter for weighting functions in the H_∞ mixed sensitivity optimization problem. The optimization problem is solved by minimizing the H_∞ norm thereby obtaining a controller that satisfies a robust performance condition. As a SISO case is considered, the obtained result can also be shown to satisfy a robust stability condition.

The main novelty is in integrating a relatively precise model of a switched Ethernet network that was developed using a modeling approach coming from the information technology research area together with a standard control theory tool.

4.4 Control of a NCS using delay estimates obtained from a network model and information on process related uncertainties

For the network performance estimate, the same network calculus based upper bound delay value is used. In the control synthesis, a more general problem is studied that also considers measurement error, as well as other perturbation sources.

A generalized model of the system is derived and a controller is constructed by minimizing the μ-value using a *D-K* synthesis procedure. The obtained controller satisfies the robust performance and robust stability conditions. Finally, in order obtain a controller more suitable for implementation, the controller is reduced to a third order controller using the Hankel singular value reduction.

Also here, the main novelty is in integrating a relatively precise model of a switched Ethernet network that was developed using a modeling approach coming from the information technology research area together with a more general control theory approach.

The main benefit of the approach is that in design it is possible to incorporate several error sources, coming from both the network as well as from the process. One drawback is that the possible conservativeness of control in the case of the upper bound condition is seldom met.

5 Control of a NCS using delay measurements

The aim of this chapter is to propose a control approach that enables a performance improvement of a control application controlled over a switched Ethernet network by utilizing the measurement of the network transmission delay. First, a method for measuring network transmission delay is presented. Second, an adaptive Smith predictor based control is proposed where the obtained network information can be utilized. Finally, the performance of the control method will be illustrated first with Matlab simulations and then with simulations on an experimental prototype.

5.1 Network transmission delay measurement

There is a clear need for instruments in the network to measure the network transmission delay. The possibility of a delay measurement enables the evaluation of the online QoS performance and its use in control. The measurement also eliminates the need for a network model, which is often difficult to obtain.

Usually, the delay measurement relies on the round trip time (RTT) measurement due to its easy implementation: no clock synchronization is required since the computations are running on the same device. However, for the NCS, RTT may not be adequate and instrumentation for measuring the end-to-end delay should be developed. The main difficulty in measuring an end-to-end transmission delay is due to timing issues like non-synchronized clocks and the scheduling policies of the operating system stack. In the switched Ethernet network, the prediction problem is further amplified due to the time-variant behavior of the transmission delay, resulting from the load of traffic variation or even the considerable changes in the network topology, especially when the network is shared with applications other than control applications.

The techniques for measuring end-to-end delay often rely on the synchronization of the clocks of the two end-systems, and sharing the same clock reference for timing events like frame sending and reception. Initially, Mills (1991) proposed a Network Time Protocol (NTP) for synchronizing the clocks of computer systems over packet-switched data networks. However, the accuracy of the NTP protocol may not be adequate for distributed systems with hard time constraints. In order to cope with the inaccuracy, a new clock synchronization procedure protocol, IEEE 1588 (IEEE 1588-2002, 2002), has recently been defined, enabling the development of one-way delay measurement methodologies for distributed systems with real time constraints.

In the IEEE 1588 standard, two kinds of clocks are defined: boundary and ordinary. Boundary clocks are used by network devices (like switches), while ordinary clocks are implemented in the devices with a single network interface card such as sensors and actuators, etc. Each network interface card with a boundary clock can act either as a master or as an ordinary clock in one network segment. The synchronization is driven by master/slaves exchanges. The time reference for this, the grandmaster clock, is chosen according to its stability and accuracy. This clock is unique for the whole architecture. The PI control algorithm is applied in reducing the error between the master clock and the slaves.

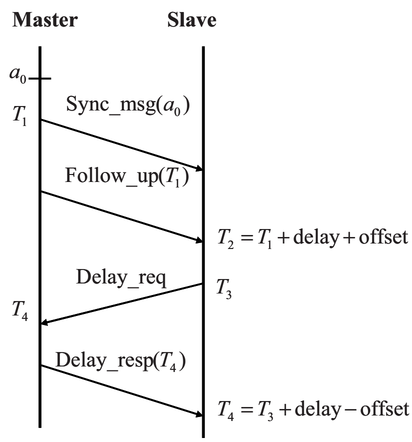

Figure 13. Transactions of PTP (Jasperneite, *et al.*, 2004).

Like in the NTP protocol, the Precision Time Protocol (PTP) of IEEE1588 is based on the $T_1 - T_4$ transaction model as shown in Figure 13. There are two stages in the synchronization process: the offset measurement stage and the propagation time measurement stage. In the offset measurement stage, the difference between the master and the slave clocks is computed. To achieve this correction, the master periodically sends a synchronization message (Sync_Msg) which contains an estimated value (a_0) of the real date (T_1) at which the message is physically sent. In the following message, entitled Follow_up, the master gives the accurate measured value of the sending date of the previous message.

In the second stage, the delay measurement enables determining the propagation time between the master and the slave devices. For this, the slave device sends a delay request message, Delay_Req, to which the master answers by the delay response message, Delay_Resp. This results in two equations, from which the delay and the offset can be solved.

$$T_2 = T_1 + \text{delay} + \text{offset} \quad (25)$$

$$T_4 = T_3 + \text{delay} - \text{offset} \quad (26)$$

As shown in Figure 13, this exchange enables measuring the propagation time with a good accuracy assuming the offset has been initially corrected. The slave is able to compute online the information necessary for its clock adjustment.

The method proposed here for the end-to-end delay measurement consists of the synchronization procedure of the network devices with the IEEE 1588 protocol, followed by frame time stamping. After the device synchronization, the frames are time stamped at the sender side and the difference between the time stamp included in the frame and the reception time is computed at the receiver side. As the frame's Ethernet header does not contain a field to add this time stamp, the information should be defined in the protocol used at the application layer. In the switched Ethernet architectures, the PTP protocol, needed for the IEEE 1588 synchronization, can easily be implemented by considering the switches as master clock references. Specific improvements in this case have been proposed by Jasperneite, *et al.* (2004) and Gardener, *et al.* (2005). The originality of the proposed approach is related to the accuracy of this estimate, especially compared to RTT measurements. In fact, this accuracy is emphasized in the context of switched topology.

Since the obtained delay measurement corresponds to the delay experienced by the last message for a given flow, this methodology only enables an estimate to be made for the latest delay. As this is based on past observations, it gives information about the trends and time-variant behavior of the delay. As a consequence of the procedure, the measurement of the delay will only be available at the receiver side. In networked control systems, this implies that the transmission delay of the last control information is stored on the actuator, even though this knowledge is more relevant to the controller. However, the information about the sensor path delay is available at the controller side, and thus can be used directly for control compensation.

5.2 Compensation of slowly varying transmission delays with an adaptive Smith predictor

5.2.1 Assumptions on the network

Prior to presenting the compensation approach, the following assumptions about the overall system are made:

1. The network transmission delay is constant, or its variation is slow.
2. The sampling period necessary to capture all relevant process dynamics is significantly smaller than the network transmission delay; in addition, the transmission delay is significant compared to the time constant.
3. After each clock cycle, the sender attempts to send a packet containing continuously updated information. After each clock cycle, the receiver examines its input queue for newly received information. If no new packet is received then the previous information is used. The sender and receiver are synchronized.
4. All control and measurement information in the network is sent in a single packet and no packet losses occur in the communication network.

The first assumption is required due to commutativity. The scheme becomes mathematically inaccurate if the transmission-delay variation is high. In a practical application, this effect may be insignificant. In fact, this is not a very restrictive assumption as in practical applications, the network transmission can be set to be constant by implementing network scheduling algorithm.

The second assumption quantifies the magnitude and relative importance of the network transmission delay with respect to process dynamics. It will be assumed that the transmission delay is significant and must be handled properly.

The third assumption states that the controller is time triggered and the receiver is event triggered.

The fourth assumption specifies the information in the packet structure and the occurrence of the transient disturbances in the network.

5.2.2 Smith predictor-based compensation scheme for a NCS

After the network transmission delay measurements have been obtained, they can be utilized with the following compensation scheme.

Consider an open loop, stable, linear process that is controlled over a network with a nominal controller. Assume that the transfer functions have been obtained and are given with the frequency domain z operator, $G(z)$, $G_c(z)$. Introduce two minor loops around nominal controller $G_c(z)$; one consisting of sensor path delay measurement $z^{-\hat{\tau}_{sc}}$, plant model $\hat{G}(z)$ and actuator path delay measurement $z^{-\hat{\tau}_{ca}}$ and another minor loop consisting of plant model $\hat{G}(z)$ as illustrated in Figure 14. Notions τ_{sc}, τ_{ca} represent the magnitude of actual transmission delay and notions $\hat{\tau}_{sc}$, $\hat{\tau}_{ca}$ represent the measured values of transmission delays.

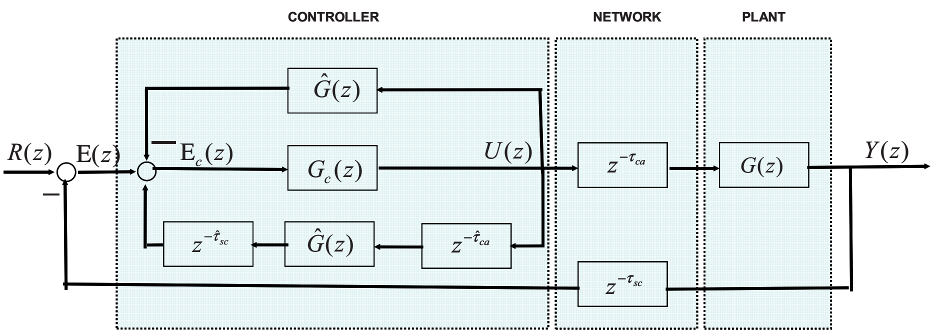

Figure 14. A Smith predictor for compensation of the network transmission delay.

With this configuration, the net result of introducing the minor loops is compensation of the time delay factor caused by the network transmission delays from the feedback signal, where it causes stability problems and moving it outside the loop where it has no deteriorating effect on closed-loop stability.

Observe that in the case where the plant model is known, and the measurement for the sensor and actuator path delays are available and equal, the actual sensor and actuator delays values τ_{sc}, τ_{ca}, the delayed process measurement and the delayed process output estimate become equivalent:

$$z^{-\tau_{sc}} \cdot G(z) \cdot z^{-\tau_{ca}} = z^{-\hat{\tau}_{sc}} \cdot \hat{G}(z) \cdot z^{-\hat{\tau}_{ca}} \tag{27}$$

and therefore the signal reaching the controller, designated as $E_c(z)$ in the diagram, is a corrected error signal given by:

$$E_c(z) = R(z) - (z^{-\tau_{sc}} \cdot G(z) \cdot z^{-\tau_{ca}} - z^{-\hat{\tau}_{sc}} \cdot \hat{G}(z) \cdot z^{-\hat{\tau}_{ca}})U(z) - \hat{G}(z)U(z) \tag{28}$$

or

$$E_c(z) = R(z) - \hat{G}(z)U(z) \tag{29}$$

This implies that in the case where the plant and the network transmission delays τ_{sc} and τ_{ca} are commutative, the error signal reaching the controller is calculated on the basis of an (open loop) predicted estimate of the process output.

With the delay measurement procedure presented in the previous section, it is only possible to get an online estimate of the measurement path delay at the controller side, and the actuator path delay measurement corresponds to the delay at the previous sampling instant. Therefore, the total compensation is not possible if the actuator delay has changed, and therefore the controller should be tuned with respect to $G(z) \cdot z^{-\tau_{ca}}$.

The main benefit of the Smith predictor scheme compared to other time delay compensators (TDC) is the online use of the delay value. This is beneficial in the NCS case, and the delay measurement can be integrated directly and modified online as the delay value changes. In several other TDC schemes, the controller is obtained as a result of optimization, which may have time constraints to be applied in the NCS.

However, there are several issues that need to be addressed when applying the predictor scheme.

The first is related to the process, as the Smith Predictor is only suitable for open loop stable processes. This can easily be seen by examining the open loop transfer function from error $E(z)$ to controller output $U(z)$:

$$\frac{U(z)}{E(z)} = \frac{G_c(z)}{1 + G_c(z)\hat{G}(z) - G_c(z) \cdot z^{-\hat{\tau}_{cs}} \hat{G}(z) \cdot z^{-\hat{\tau}_{cs}}} = \frac{G_c(z)}{1 + G_c(z)\hat{G}(z)(1 - z^{-\hat{\tau}_{cs}} z^{-\hat{\tau}_{cs}})} \tag{30}$$

If $\widehat{G}(z)$ has an unstable pole, this will have an unstable zero, and therefore the closed loop becomes unstable as there is an unstable pole-zero cancellation.

Another issue is related to uncertainties in the process model - the structure in Figure 14 is sensitive to process model errors and should thus be taken into account when using the scheme. To study the effect of plant model and measurement uncertainties, consider Figure 15 which has been obtained by rearranging the blocks of Figure 14 to better illustrate the effect of model errors.

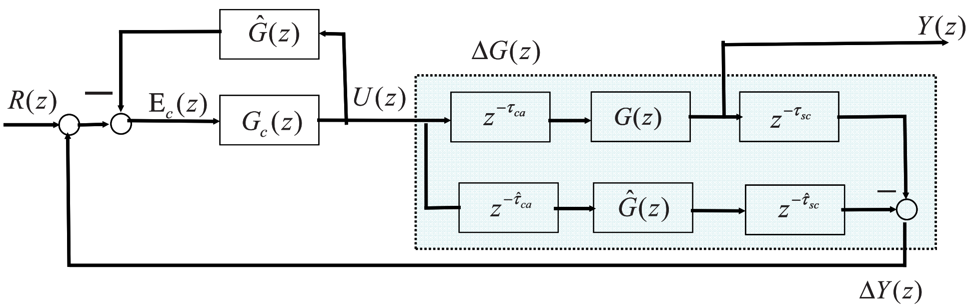

Figure 15. Uncertainty system resulting from a model mismatch

According to the small gain theorem, the closed loop multivariable system is stable if its stable loop transfer function $L(z)$ satisfies:

$$\|L(z)\| = \|L(e^{-i\omega})\| < 1 \qquad \forall\omega \tag{31}$$

where $\|.\|$ denotes any norm satisfying $\|AB\| < \|A\| \cdot \|B\|$, and ω is angular frequency. For the system in Figure 15, the system is stable if:

$$\|L(z)\| = \left\| \frac{G_c(z)}{1 + G_c(z)\hat{G}(z)} \cdot \Delta G(z) \right\| < \left\| \frac{G_c(z)}{1 + G_c(z)\hat{G}(z)} \right\| \cdot \|\Delta G(z)\| < 1 \qquad \forall\omega \tag{32}$$

From this it can be seen that for sufficiently small plant uncertainties, stability can be guaranteed. However, in the face of small delay measurement errors, which are inevitable in the NCS, the Smith predictor may give poor performance as the error is amplified by the high controller gain.

For this reason, it is thus imperative in applications of the Smith predictor compensation for networked control to apply an adaptive procedure. One commonly used approach is to update the model parameters as the process parameters vary in order to minimize $\|\Delta G(z)\|$. In practice, this may lead to closed loop instability as a result of the model parameter mismatch. Furthermore, the adaptation procedure may not be suitable for the NCS environment due to timing constraints.

In this work a different adaptation procedure is proposed which is expected to fit better to the NCS environment. In order to ensure the robust stability and performance of the system, the adaptation will be achieved by changing the nominal controller parameters according to the changes in the delay value error. In the case where the delay error increases significantly, the controller gain is reduced temporarily in order to satisfy the robust stability and performance conditions. This will obviously reduce the value of $\left\|\dfrac{G_c(z)}{1+G_c(z)\hat{G}(z)}\right\|$.

As a limiting criterion that is verified continuously, a modification of the criteria presented in Morari and Zafiriou (1989) will be used. According to Morary and Zafiriou, the necessary and sufficient conditions for first order systems with deadtime for PI controller gain k_{new}, including the Smith predictor, to remain robustly stable in the presence of a small mismatch in delay is:

$$k_{new} = \frac{\tau}{\bar{\delta}k} \tag{33}$$

where k is process gain, τ is process time constant, and $\bar{\delta}$ is maximum delay error. In this work, the criterion will be modified according to the dominant process time constant.

A final note is related to the commutativity. In the case when the commutativity assumption for the linear system and network transmission delay does not hold, or

$G(z)\cdot z^{-\tau_{ca}} \neq z^{-\tau_{ca}} G(z)$, this would imply that grouping the time-variant delays and grouping the controller and linear time-invariant system together is not possible and thus more advanced methods should be used.

5.3 Simulation study for the Smith predictor scheme

The effectiveness of the control approach are illustrated here and evaluated through an example with Matlab simulations. First, the results of a simulation study are presented where it is assumed that there are no errors in measurements. Next, the effect of measurement errors on the performance of the compensation scheme is studied.

5.3.1 Simulations with no measurement errors

Suppose that the sensors, actuators and controller of the plant are spatially distributed and connected over the communication network. In continuous time, the process model of the real-time process and a nominal controller (time in ms) are described as follows:

$$P(s) = \frac{2}{(0.2s+1)(5s+1)} = \frac{2}{s^2+5.2s+1} \tag{34}$$

$$C(s) = \frac{K_P s + K_I}{s}, \qquad K_P = 1.7445, \quad K_I = 0.2224 \tag{35}$$

The controller parameters for the nominal controller were obtained by minimizing the integral of square errors (ISE) for the system with a constant network transmission delay of 0.5 ms in the actuator and sensor paths.

A discrete version of the continuous system was used in the simulations where 0.02 ms was used as a sampling period. The transmission delays over the network in the sensor and on the actuator side were assumed to vary randomly between zero to some upper delay value. The variation of the delay values followed the uniformly distributed random signal, which changed with a frequency of 200 times the sampling period. Four cases were simulated with different upper bound delay values, starting from the shortest delays with an upper bound of 0.5 ms, and increasing the upper bound delay to a value of 3.5 ms.

The step response of the PI controlled discrete closed loop system, together with the simulated delays in the sensor and actuator paths, is presented in Figure 16. The first column of the figure illustrates the simulated delay variation in the sensor part; the second column the delay variation in the actuator path; and the third the response of the closed loop system. In the first row, the performance of a nominal PI controller under design specification is presented and the figures on rows 2-5 illustrate the performance under increased delay conditions. According to the figure, the nominal controller is unable to maintain the performance of the system. Even though the performance is acceptable at the lowest delay value, the performance significantly decreases as the network transmission delay values increase and, finally, even approaches a stability limit at the highest delay values. Therefore, it can be concluded that a compensation procedure is required.

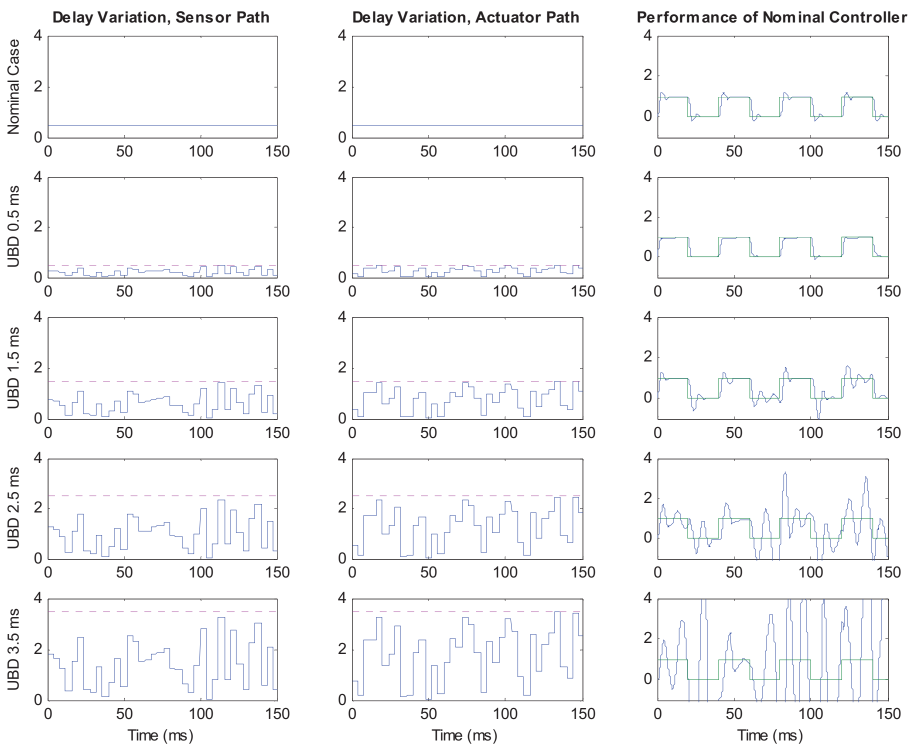

Figure 16. Performance of the nominal (IAE optimized) PI controller under increased delay variations conditions.

Figure 17 presents the case in which the Smith predictor based compensation scheme is applied with the same nominal controller. It was assumed in the simulations that the model of the rational part of the process is known, and that the only uncertainty arises from the delay measurement error resulting from the arrival of the measurement for the actuator delay one sample later. The same sampling period as in the simulations for the nominal controller was used. The uncertainty related to the measurement error was simulated by introducing an additional dead time for arrival of the network transmission delay measurement in the Smith predictor minor loop. In the first two columns of the figure, the actual actuator and the sensor delay values are illustrated together with the respective measurement values. The close overlapping of the graphs illustrates the accuracy of the measurement and, therefore, in this case the adaptation action was not active. The third column represents the response of the closed loop system. It can be concluded from the figure that the performance of the system is improved significantly and that the control becomes less sensitive to the delay effect produced by the network.

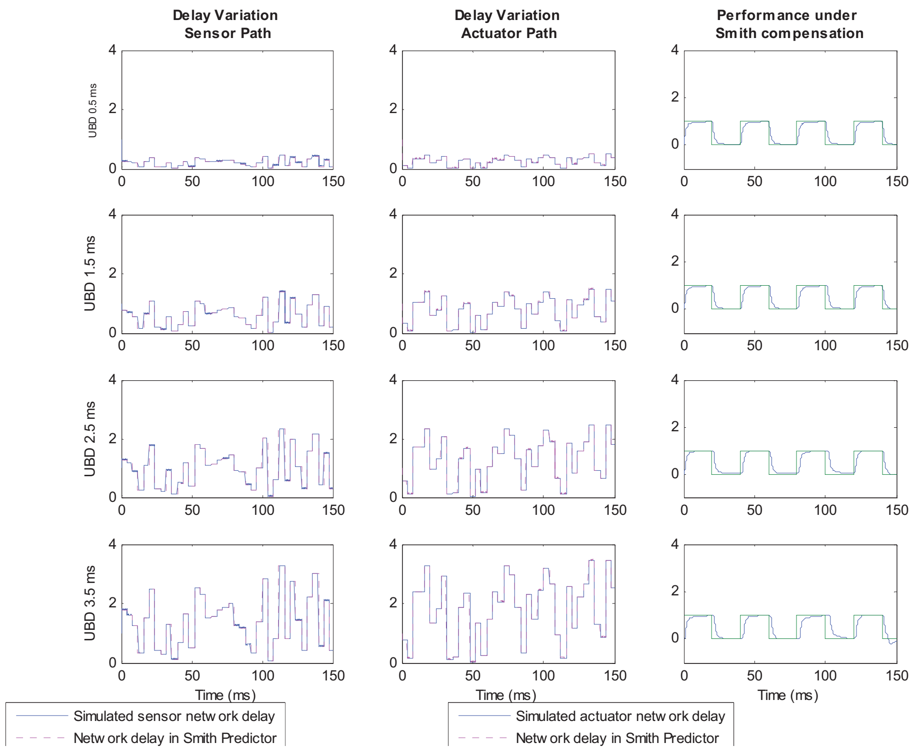

Figure 17. The performance of the Smith predictor compensation scheme when the delay varies between zero and the upper bound delay.

5.3.2 Simulations with measurement errors

Even though the one-way delay measurement is available, it often includes several sources of uncertainty that affect its usability in several time delay compensator schemes. In Section 5.1 for example, a one-way delay measurement was achieved due to the clock synchronisation protocol. In order to take into account the clock's derivation (like clock drift), and thereby to provide accurate delay measurements, it is necessary to synchronize the clocks by periodically sending the messages, which increases the network load and hence the delays. Moreover, the protocol assumed that the delay variation remained limited between two synchronization steps. The technique can also only be used in point-to-point topologies (like switched Ethernet) and has to be implemented in all communication interfaces of the controllers, sensors, actuators and network devices. In addition, the purpose of the IEEE 1558 protocol is to ensure clock synchronization between the different communication nodes, but not to measure the delays. This means that it was necessary to embed delay measurement algorithms inside the sensors/actuators and to transport delay measurements to the controller. The approach proposed also requires smart instruments able to support additional software for measuring delays that limit its general usage.

The influence of the delay measurement error on the control performance for the Smith predictor scheme is next studied using the Matlab simulations. Let first consider the adaptation procedure that is going to be utilized. The adaptation consists of adapting the gain of the controller according to the delay measurement error. If, due to the error, the gain rises over the value k_{new} calculated from equation (33), the controller gain is temporarily reduced. Consider Figure 18 where three simulation results are illustrated: in the case when there is no error in the measurement and the gain of PI controller in Smith predictor scheme is 10 times the gain of nominal controller; the case when there is 10% error in the delay measurement and the gain of the controller in 10 times higher the gain of the nominal PI controller (higher than k_{new}); and the case when the error is 10% and the gain has been reduced to 5 times the nominal controller to be equal to the value of k_{new}. For each case, the figure shows both frequency and step response. From the frequency and step responses of the first row, an obvious conclusion can be drawn: if there is no error in the delay measurement, the Smith predictor scheme provides a much higher bandwidth for control and thus a faster response is possible. From the second row, it can be concluded that when a 10% error in delay measurement is introduced, and the gain of the controller in Smith predictor remains 10 times higher than in the nominal case (or twice the value of k_{new}), the frequency response of a closed loop transfer function becomes significantly different from the nominal case and also the time domain performance significantly gets worse. Finally, the third row shows that it is possible to regain some of the performance as the value is reduced to the value k_{new}.

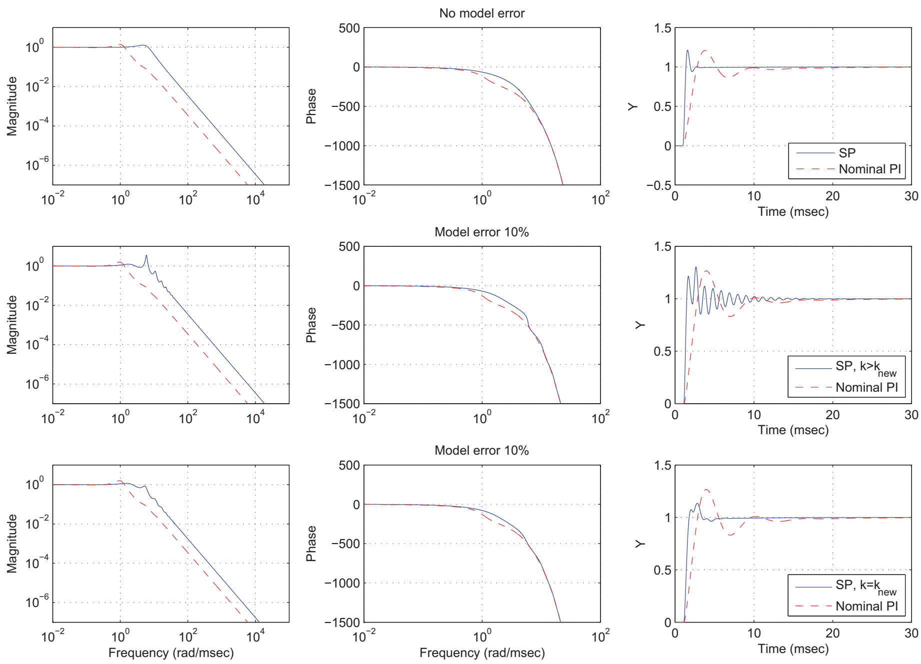

Figure 18. The effect of gain adaptation on the response of the system.

The simulation study for the NCS was performed as follows. The maximum measurement error in the delay utilized in the Smith predictor schemes was gradually increased and the effect on the process performance studied under different delay conditions. Four delay measurement error conditions are evaluated, the error being increased from 25% to 100%. The case where the maximum error is 100% corresponds to the condition where the measurement used in the Smith predictor is possibly two times higher than the actual network transmission delay. This effect of error increase was studied under two delay conditions of different magnitude: in the first case, the delay value varied between 0 and 2.5 milliseconds; in the second, it was between 0 ms and 3.5 ms.

The results of this study are presented in Figure 19 and Figure 20. Figure 19 presents the performance of the Smith predictor and the adaptive Smith predictor when the upper bound delay is 2.5 ms. The first column in the figure represents the sensor network delay and the delay used in the Smith predictor schemes; the second column shows the actuator network delay and the delay used in the Smith predictor schemes; columns 3 and 4 illustrate the process performance under the Smith predictor scheme and adaptive Smith predictor schemes. Figure 20 presents a similar increase in maximum error, with a different upper bound delay value of 3.5 milliseconds.

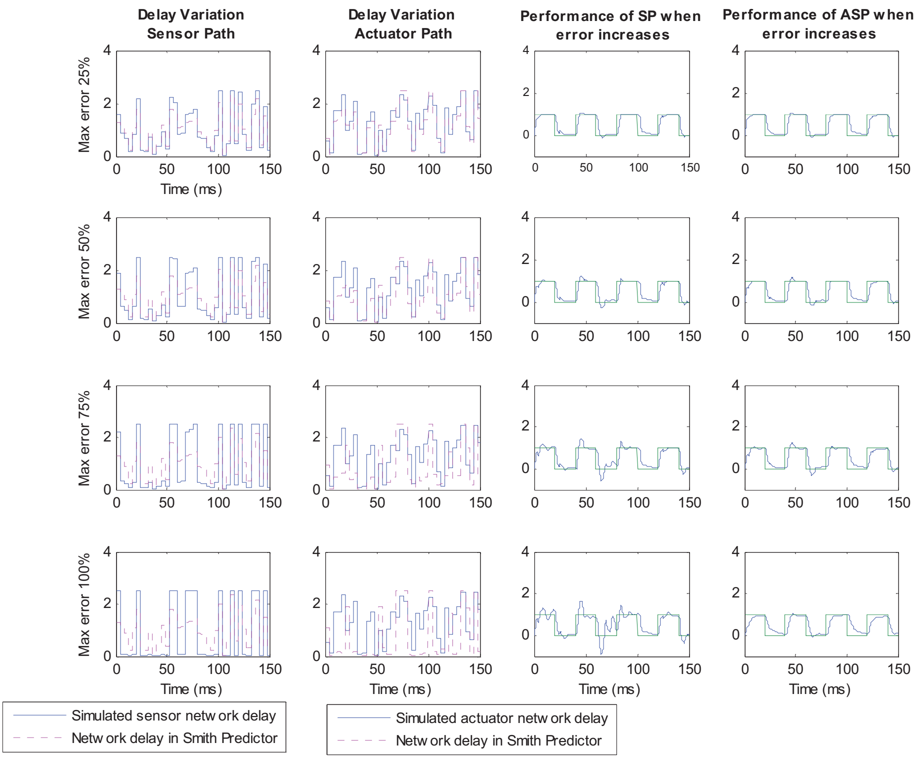

Figure 19. Performance of the Smith predictor and adaptive Smith predictor approaches as the delay value increases. UBD = 2.5 ms.

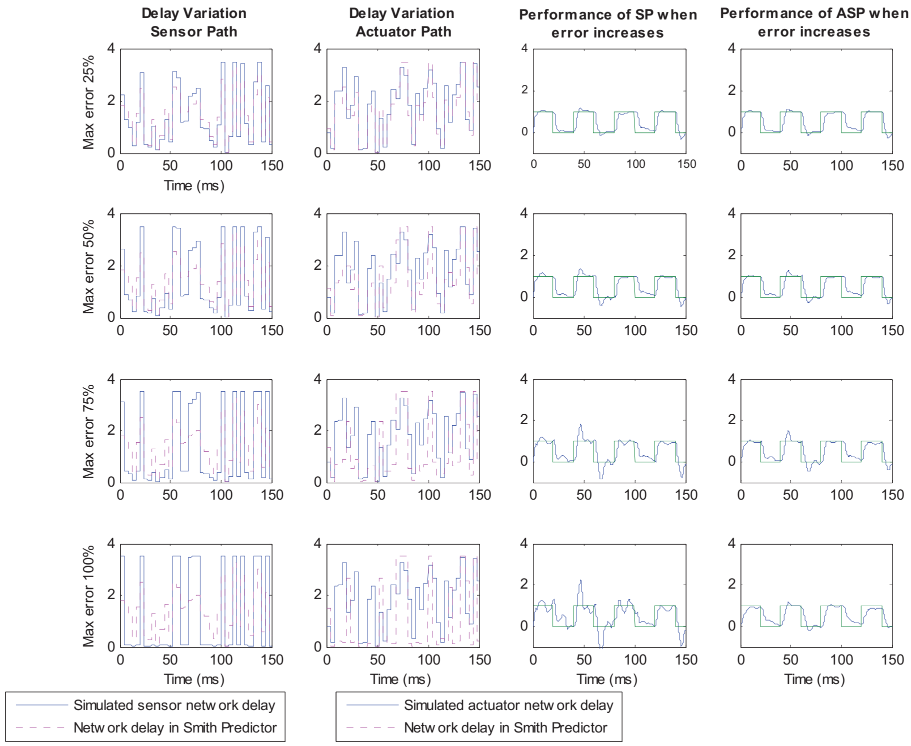

Figure 20. Performance of the Smith predictor and adaptive Smith predictor approaches as the delay value increases. UBD = 3.5 ms.

It can be concluded from the figures that as the error in the delay measurement increases, the performance of the Smith predictor based approaches is significantly reduced and at some point it becomes unacceptable. Using the adaptation procedure it is possible to regain some of the performance, although the adaptation strategy at some point also becomes inadequate. In Figure 21, the situation is illustrated even more clearly by presenting the extreme case in which the upper bound delay is used as a delay estimate. As the upper bound delay value increases from 0.5 to 3.5 due to the resulting model mismatch, the performance of the control system decreases and the system approaches the stability limit when the upper bound delay value equals 3.5.

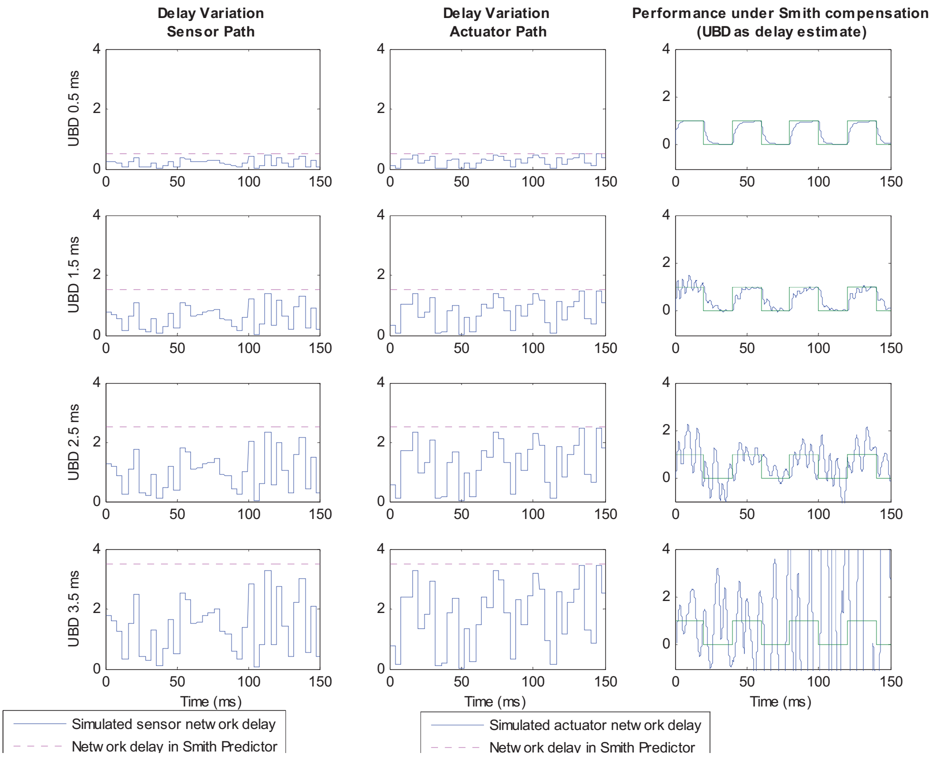

Figure 21. Performance of the Smith predictor when the end-to-end upper bound estimate is used as a delay estimate in the Smith predictor scheme.

5.3.3 Testing the Smith predictor approach in the experimental platform

A platform for the behavioral analysis of the NCSs was developed for studying the effects of the communication network on the performance of the control system. Physically, the platform consists of the communication network and two computers dedicated to simulate the real time process and the controller, respectively. The Ethernet switches (Cisco Catalyst 2912 XL) are augmented with additional embedded intelligence in order to cope with real-time and fault tolerance requirements. In order to cope with real time requirements, the sampled versions of the process model and the controller were directly programmed using C- language and, to further reduce the calculation period, real time programming was necessary and performed through the RealTime Application Interface (RTAI), which enables writing applications with strict timing constraints for the Linux operating system.

In the simulation of the process and controller, three different kinds of threads were defined. The first thread type is responsible for computing a new control action (or the output of the process). Since computation of the control action has the hardest real time constraints, the management of the communication network cannot be achieved by this thread only. Two other thread types have been developed to handle this problem: one thread type to copy and to send the information delivered by the first thread type to the second computer, and a second thread type to retrieve the information from the packet sent by the second computer and to forward it to the first thread. In the experimental setup, the clock cycle was chosen to be shorter than the sampling period of the system. The information exchanged between the threads on one side is managed by a shared memory, where every buffer is monitored and locked by the thread that accesses it. Maximum priority is given to all these three threads compared to the other tasks on the computer in order to respect the hard time constraints. The FIFO scheduling policy was selected on the system as presented in the POSIX1 standard.

The application protocol defined for the communication is based on the network stack Ethernet/IP/UDP (losses are not managed in this case by the transport protocol). Packets sent contain an identifier of the communication, a packet number, the value exchanged by the computers, a time stamp and a value of the last delay. These last two fields are necessary for measuring the one way delay.

In order to achieve the clock synchronization of the two PCs, a PTP daemon is implemented on each computer. The synchronization is performed online in parallel with the control computation and packet transmission. As presented in Section 2.1, the controller is only aware of the delay from the sensors to the controller, but not of the delay from the controller to the process, which is only stored on the process side. For this reason, a field that corresponds to the last delay measurement stored in the process was included into the packet transferred to the controller, i.e. the controller will receive this information and will be able to use it for the Smith predictor.

The platform was used to test the performance of the Smith predictor when the delay estimate was modified online using the result from the IEEE-based, one-way delay algorithm. The discrete controllers and the process were implemented in C-code and simulated on the platform. In the experimental prototype, the dynamics were slowed down by a factor of ten due to computational issues. However, because the scaling has been performed also in the network part, the performance of the platform well reflected the actual situation.

The simulation result from the experimental prototype is presented in Figure 22. Four graphs are presented: the first two illustrate the Smith predictor performance when there is no load in the network and the round trip time has a low value, and the last two show the performance when a load is introduced into the network that raises the RTT value so that it is upper bounded by 30 ms. The first two graphs correspond to the Matlab simulation result presented in the first row of Figure 16; the last two correspond to the Matlab simulation result presented in the second row of Figure 17. This can be seen by noticing that the round trip time experienced by the controller is upper bounded by 30 ms, which corresponds to the upper bound of 15 ms for the sensor and actuator path delays, or to the 1.5 ms that was used in the Matlab simulation.

It can be concluded from the figures that the performance obtained with the experimental prototype corresponds to that obtained with the Matlab simulations. It can also be concluded that the combined Smith predictor and the delay measurement algorithm is able to maintain the performance level of the system. As can be seen in the figure, the response is not always identical to the response in Figure 17 for every step. The main reason for the difference was due to the network delay difference between the Matlab simulations and the simulations on the experimental prototype. In fact, the form of the responses becomes more similar as the delay variation in both cases is reduced, and becomes identical if the delay is set constant.

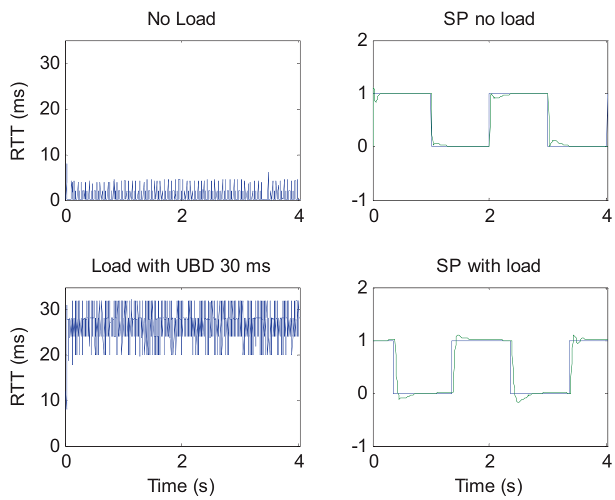

Figure 22. Performance of the Smith predictor with the online delay parameter under a varying network transmission delay.

6 Time-variant control method for a NCS

This chapter proposes an approach for networked control systems that enables the performance improvement of a control application by utilizing the measurement of the network transmission delay. The assumptions concerning the network are otherwise the same as those presented in Section 5.2.1, with the difference that the network transmission delay is now time-varying. The networked control system will be modeled as a time-varying difference system, where the time-varying behavior comes from the communication network only, and the process will be considered time-invariant.

Controller and observer design for linear time-varying systems is generally a more difficult problem because of the fundamental problems related to the analysis of such systems. The classical theory does not provide much help because of many concepts that do not carry over to time-varying system, such as commutativity of subsystems, the concept of poles and zeros, global controllability and observability, etc. Generally, time variant systems cannot be solved in terms of elementary functions, and solutions may exhibit a finite escape time. Further, frequency domain methods are not applicable and state space based approaches to delayed systems become more complicated as the dimensions of the state become time-varying and there are no sound mathematical tools to handle this problem.

In this chapter the polynomial systems theory will be applied to controller and observer design for a time-varying NCS. The theory is well established for time-invariant differential and difference linear systems, (Blomberg and Ylinen, 1983). The main idea behind the theory is that it utilizes the algebraic properties of polynomials with real or complex coefficients. The theory is computational in nature, and it uses the fact that the ring of polynomials over a field in an operator normally satisfies a division algorithm which can be used to find common factors and to manipulate polynomial matrices into suitable canonical forms in an algorithmic way. For a time-varying case, the resulting theory is similar to the time-invariant one, (Ylinen, 1980). This is one of the key benefits compared to several other theories on time-varying systems. The main difference to the time invariant case is that in time-varying case the algebraic methodology related to noncommutative skew polynomials is needed. As parameters of time-varying systems are functions of time, all computations must be done symbolically. This limits the complexity of the problems that can be handled as no numerical methods can be used.

This chapter is organized as follows. First an example is given that illustrates the noncommutativity of time variant systems; next, the polynomial systems theory is applied for compensation of time-varying delays; for this, the process and the network is first presented as a composition of time-varying difference input-output (IO-) relations. Second, the time-varying delay is presented as noncommutative skew polynomials. Third, an overall IO-relation for the network and process is derived. Forth, the obtained overall IO relation is analyzed. Finally, in Sections 6.2.5, observer design and feedback control problems are formulated. The chapter ends with illustrations of proposed observer and feedback control approaches with simulations.

6.1 On commutativity of time-varying linear systems

In this subsection, the commutativity of the time-varying linear discrete systems is studied. The aim is to determine whether it is possible to interchange two different time-variant systems and obtain the same equivalent system for both cases.

Consider the commutativity of two discrete linear systems in series, see Figure 23. In the figure, $\tau_{sc}(k)$ and $\tau_{ca}(k)$ are time variant discrete delays, $x(k)$ and $u(k)$ are input sets and $v(k)$ is the output set. Consider the SISO case where only one variable is sent through the system.

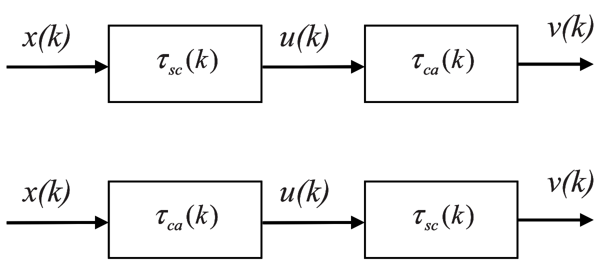

Figure 23. Two time-varying linear difference systems with blocks interchanged.

By simple inspection we can write for the upper system:

$$u(k) = x(k - \tau_{sc}(k)) \tag{36}$$

$$v(k) = u(k - \tau_{ca}(k)) \tag{37}$$

Combining two equations:

$$v(k) = x(k - \tau_{ca}(k) - \tau_{sc}(k - \tau_{ca}(k))) \tag{38}$$

For the system in the lower part of Figure 23 we can write as follows:

$$u(k) = x(k - \tau_{ca}(k)) \tag{39}$$

$$v(k) = u(k - \tau_{sc}(k)) \tag{40}$$

$$v(k) = x(k - \tau_{sc}(k) - \tau_{ca}(k - \tau_{sc}(k))) \tag{41}$$

Comparing the two equations obtained for $v(k)$ it can be seen that generally, the commutativity does not hold. It can also be concluded that the commutativity holds only under the following conditions:

$$\tau_{ca}(k) + \tau_{sc}(k - \tau_{ca}(k)) = \tau_{ca}(k - \tau_{sc}(k)) + \tau_{sc}(k) \tag{42}$$

This is satisfied when the time-variance is the same

$$\tau_{ca}(k) = \tau_{sc}(k) \qquad \forall k \tag{43}$$

or in particular when the delays are constant

$$\tau_{sc}(k) = \tau_{sc} \text{ and } \tau_{sa}(k) = \tau_{sa} \qquad \forall k \ . \tag{44}$$

The condition in (43) is valid only for these special simple systems; a more general rule is (44).

6.2 Polynomial systems theory applied to compensation of time-varying delays in networked control systems

6.2.1 Presentation of a NCS as a composition of time-varying difference IO relations

In this section the process and the network of NCS are presented as a composition of time-varying difference input-output (IO-) relations. Consider a time-varying discrete time input-output system by difference models of the form:

$$\sum_{i=0}^{n} a_i(k)y(k-i) = \sum_{i=0}^{m} b_i(k)u(k-i) \tag{45}$$

where $u, y \in X$ are real or complex-valued input and output signals on a time set $\mathbb{Z}$, and a_i, b_i are coefficients from a suitable space K of real or complex-valued functions on $\mathbb{Z}$.

Let us define unit delay operator r as

$$(rx)(k) = x(k-1) \tag{46}$$

and unit predictor operator q as

$$(qx)(k) = x(k+1) \tag{47}$$

Then Equation (45) can be written shortly as an operator equation

$$\underbrace{\sum a_i r^i}_{a(r)} y = \underbrace{\sum b_i r^i}_{b(r)} u \Rightarrow a(r)y = b(r)u \tag{48}$$

or alternatively with the unit predictor operator by

$$\underbrace{\sum \tilde{a}_i q^i}_{\tilde{a}(q)} y = \underbrace{\sum \tilde{b}_i q^i}_{\tilde{b}(q)} u \Rightarrow \tilde{a}(q)y = \tilde{b}(q)u \tag{49}$$

A set of linear, time-varying linear operator equations can be written as matrix equations:

$$A(r)y = B(r)u \Leftrightarrow [A(r) \vdots - B(r)]\begin{bmatrix} y \\ u \end{bmatrix} = 0 \tag{50}$$

where $y \in X^q$, $u \in X^r$, and $A(r)$ and $B(r)$ are polynomial matrices of sizes s by q and s by r. A multivariable input-output relation generated by Equation (50) can be defined as a set

$$S = \left\{ (u, y) \mid [A(r) \vdots - B(r)]\begin{bmatrix} y \\ u \end{bmatrix} = 0 \right\} \tag{51}$$

The matrix $[A(r) \vdots - B(r)]$ is called also a generator for S. Generators for the same input-output relation are input-output (IO-) equivalent. A composition of input-output relations consists of a set of input-output relations and a description of the interconnections between the signals. From a composition it is always possible to write an internal IO relation S_i, the output of which consists of the outputs of all IO-relations and then interconnections are taken into account.

For the NCS if the process is described by difference equation of the form:

$$A_p(r)x = B_p(r)v, \tag{52}$$

the controller to actuator transmission delay by time-varying difference equation:

$$v = D(r)u, \tag{53}$$

and the sensor to controller transmission delay by time-varying difference equation:

$$y = F(r)x, \tag{54}$$

the following IO- relations can be defined:

$$S_p = \left\{ (v, x) \mid [A_p(r) \vdots - B_p(r)]\begin{bmatrix} x \\ v \end{bmatrix} = 0 \right\} \tag{55}$$

$$S_a = \left\{ (u, v) \mid [I \vdots - D(r)]\begin{bmatrix} v \\ u \end{bmatrix} = 0 \right\} \tag{56}$$

$$S_s = \left\{ (x,y) \mid [I \vdots -F(r)] \begin{bmatrix} y \\ x \end{bmatrix} = 0 \right\} \tag{57}$$

where $x, y \in X^q$, $u, v \in X^r$, and $D(r)$ and $F(r)$ are polynomial matrices of sizes r by r and q by q. $A_p(r), B_p(r)$ are polynomial matrices of sizes s by q and s by r.

Furthermore, taking into account the connections between the IO-relations as they are in the networked control system, the composition of IO-relations as presented in Figure 24 can be constructed. The composition can be written formally as an internal IO-relation S_i:

$$S_i = \left\{ (u,v,x,y) \mid \begin{bmatrix} I & 0 & 0 & \vdots & -D(r) \\ -B_P(r) & A_P(r) & 0 & \vdots & 0 \\ 0 & -F(r) & I & \vdots & 0 \end{bmatrix} \begin{bmatrix} v \\ x \\ y \\ u \end{bmatrix} = 0 \right\} \tag{58}$$

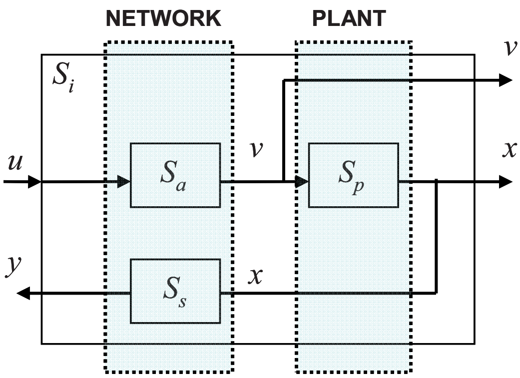

Figure 24. Composition of IO-relations for a NCS.

6.2.2 Presentation of time-varying delay with skew polynomials

The aim of this section is to derive the representation for the time-varying delays terms $D(r)$ and $F(r)$ of the Equations (56) and (57). The representation will be given as non-commutative skew polynomial equations. The background on skew polynomials can be found in Cohn (1971).

Most of the concepts and properties of ordinary polynomials can be generalized to skew polynomials. As a separation from the set of the ordinary polynomials which form a commutative ring with coefficients in K, the skew polynomials can be defined as a noncommutative ring with coefficient space K as a subring. The sum of two skew polynomials is defined as for ordinary polynomials. The product is more complicated, as the multiplication of these time-varying polynomials defined by composition of operators is not commutative. The product of two skew polynomials can be constructed using the following property repeatedly:

$$((rb)x)(k)=(r(bx))(k)=(bx)(k-1)=b(k-1)x(k-1)=(r_K b)(k)(rx)(k)= \\ ((r_K b)(rx))(k) \tag{59}$$

where $b \in K$, r_K is the unit delay operator on the coefficient space K. The degree and uniqueness of description are satisfied as for ordinary polynomials if the signal space is sufficiently rich. The choice that X is the set of complex-valued signals on a time set $\mathbb{Z}$ guarantees this.

Division algorithms such as the (right) division algorithm (RDA) are satisfied uniquely if and only if the coefficient ring K is a field. Unfortunately, the most time-varying coefficient rings do not satisfy this, which means that they must be extended to rings of rational functions. This is especially important, because the division algorithms are needed for the manipulation of the skew polynomial matrices used in the descriptions of multivariable systems. A skew polynomial can be invertible as a skew polynomial only if it is of degree zero. However, there can also be skew polynomials of a higher degree which are invertible as mappings (e.g. r and q). The main properties of skew polynomials are summarized more formally in Table 3.

Table 3. Properties of skew polynomials.

Sum:	The sum of two skew polynomial operators can be written as: $a(r)+b(r)=\sum a_i r^i + \sum b_i r^i = \sum (a_i+b_i) r^i$
Product:	The product $(\sum a_i r^i)(\sum b_i r^i)=\sum c_i r^i$ must satisfy $rb = r_K(b)r + 0_K$ where $b \in K$, r_K is the unit delay operator on coefficient space K, and 0_K is the zero operator on K.
Degree	The degree of skew polynomial is well-defined if the signal space X is sufficiently rich to make the powers $r^0, r^1, r^2, \ldots$ linearly independent.
Uniqueness	The skew polynomial is well-defined if the signal space X is sufficiently rich to make the powers $r^0, r^1, r^2, \ldots$ linearly independent.
Division algorithms:	The division algorithms are satisfied uniquely for all $a(r), b(r) \neq 0$ if, and only if, the coefficient ring K is a field, e.g. the (right) division algorithm (RDA): $a(r)=b(r)c(r)+d(r), \qquad \deg d(r) < \deg b(r)$
Invertibility	The skew polynomial can be invertible as a skew polynomial only if it is of degree zero. There can also be skew polynomials of a higher degree which are invertible as mappings (e.g. *r* and *q*).

Table 4. Properties of skew polynomial matrices.

Unimodularity	Skew polynomial matrix is unimodular if it is invertible as a skew polynomial matrix.
Determinant	For skew polynomial matrices there is no determinant which could be used for testing the unimodularity.
Invertibility	A skew polynomial matrix can be invertible as a mapping even though it is not unimodular.
CUT form	Skew polynomial matrices can be brought to row or column equivalent forms e.g. to an upper triangular form using the elementary operations based on division algorithm, see (Section 6.2.3.)
Row (Column) equivalence	Two skew polynomial matrices $A(r), B(r)$ are row (column) equivalent if there is a unimodular matrix $P(r)$ so that $A(r) = P(r)B(r)$, $(A(r) = B(r)P(r))$

For the MIMO case, a matrix with skew polynomial entries, the skew polynomial matrix should be defined. The properties of skew polynomial matrices are summarized more formally in Table 4.

Provided that the time-varying controller to plant transmission delay is a multiple of the sampling period, it can be presented as

$$v(k) = u(k - \theta(k)) = (r^{\theta(.)}u)(k) \tag{60}$$

Representing this as a skew polynomial representation, the following skew polynomial equation can be written:

$$v = (d_0 + d_1 r + \ldots + d_n r^n)u = d(r)u \tag{61}$$

where the coefficients are zero otherwise but

$$d_{\theta(k)}(k) = 1 \tag{62}$$

Another way to describe the time-varying delay is to use prediction:

$$v(k + \vartheta(k)) = (q^{\vartheta(.)}v)(k) = u(k) \tag{63}$$

which can be written as a skew polynomial equation

$$(\tilde{c}_0 + \tilde{c}_1 q + \ldots + \tilde{c}_n q^n)v = \tilde{c}(q)v = u \tag{64}$$

This is more complicated description, because it is not necessary a mapping $u \mapsto v$. If the delay θ and the prediction interval ϑ are related to each other by

$$\vartheta(k) = \theta(k + \vartheta(k)) \tag{65}$$

then

$$q^{\vartheta(.)} r^{\theta(.)} u = u \tag{66}$$

for all $u \in X$ and vice versa. This means that all values of $u(k)$, $k \in \mathbb{Z}$, are used, or more formally, for each $k \in \mathbb{Z}$ there exists $l \in \mathbb{Z}$ such that $v(l) = u(k)$. Correspondingly

$$\theta(k) = \vartheta(k - \theta(k)) \tag{67}$$

if and only if

$$r^{\theta(.)} q^{\vartheta(.)} v = v \tag{68}$$

for all $v \in X$. This means that no value $u(k)$ is used more than once. Thus $r^{\theta(.)}$ and $q^{\vartheta(.)}$ as well as the corresponding skew polynomials are invertible mappings if (66) and (68) are satisfied. This means that each value $u(k)$ appears exactly once in the values $v(k)$. Equations (65) and (67) are needed when interchanging the description of delay between (61) and (64). Similarly to the controller and to the actuator transmission delay above, the output delay can be written in two different ways

$$y = (f_0 + f_1 r + \ldots + f_n r^n) x = f(r) x \tag{69}$$

$$(\tilde{e}_0 + \tilde{e}_1 q + \ldots + \tilde{e}_n q^n) y = \tilde{e}(q) y = x \tag{70}$$

6.2.3 General NCS composition

For analysis and control design purposes, an internal IO-relation S_i should be brought to the overall IO-relation form S_o, as illustrated in Figure 25, and described more formally as:

$$S_o = \{(u_o, y_o) \mid \exists y_1[(u_o,(y_1,y_o)) \in S_i]\}$$

It can be shown that for every S_i this is possible, if division algorithms can be applied. The mathematical machinery that is used to achieve this are elementary operations:

(i) add a row (column) multiplied from the left (right) by a skew polynomial to another row (column),

(ii) interchange two rows (columns)

(iii) multiply a row (column) from the left (right) by an invertible skew polynomial.

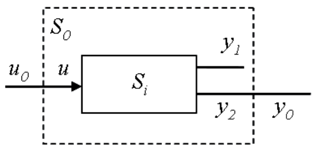

Figure 25. General composition.

Let us consider an internal IO-relation S_i obtained for networked system, Equation (58), and rewrite that for SISO case:

$$S_i = \left\{ (u,v,x,y) \,\middle|\, \begin{bmatrix} 1 & 0 & 0 & \vdots & -d(r) \\ -b(r) & a(r) & 0 & \vdots & 0 \\ 0 & -f(r) & I & \vdots & 0 \end{bmatrix} \begin{bmatrix} v \\ x \\ y \\ u \end{bmatrix} = 0 \right\} \tag{71}$$

Using elementary row operations this can be brought to the form

$$\begin{bmatrix} 1 & 0 & 0 & \vdots & -d(r) \\ 0 & a(r) & 0 & \vdots & -b(r)d(r) \\ 0 & -f(r) & 1 & \vdots & 0 \end{bmatrix} \tag{72}$$

so that v can be uniquely eliminated. Then the remaining model is written using the operator q

$$\begin{bmatrix} \tilde{a}(q) & 0 & \vdots & -\tilde{b}_d(q) \\ -1 & \tilde{e}(q) & \vdots & 0 \end{bmatrix} \tag{73}$$

where $\tilde{b}_d(q)$ is $b(r)d(r)$ rewritten with the q operator. Using again the elementary row operations gives

$$\begin{bmatrix} 1 & -\tilde{e}(q) & \vdots & 0 \\ 0 & \tilde{a}(q)\tilde{e}(q) & \vdots & -\tilde{b}_d(q) \end{bmatrix} \tag{74}$$

Thus x can be uniquely eliminated and the overall system is generated by

$$\begin{bmatrix} \tilde{a}(q)\tilde{e}(q) & \vdots & -\tilde{b}_d(q) \end{bmatrix} \tag{75}$$

This is now an overall IO relation

$$S_o = \left\{ (u_0, y_0) \mid \begin{bmatrix} \tilde{a}(q)\tilde{e}(q) \vdots -\tilde{b}_d(q) \end{bmatrix} \begin{bmatrix} y_0 \\ u_0 \end{bmatrix} = 0 \right\} \tag{76}$$

as depicted in Figure 26, and this may be used as a starting point for the analysis, observer and feedback controller design for the NCS problem.

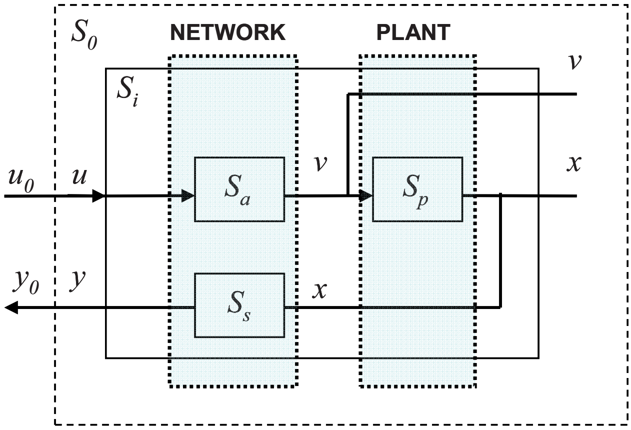

Figure 26. Overall IO-relation of a NCS.

6.2.4 Analysis of an overall IO-relation of a networked system

The overall IO-relation obtained can be used for analysis as such. Here the realizability, stability, controllability and observability of the IO-relation are studied.

Realizability. The realizability of a system model requires that the output of the model can be solved uniquely whenever the input and a sufficient, finite number of initial values of the output are given. The model is then said to be regular. If only past and present values of the input are needed for solving the output, then the model is nonanticipative or causal. For instance, an overall IO-relation for NCS (76) is causal if the degree of $\tilde{a}(q)\tilde{e}(q)$ is not lower than the degree of $-\tilde{b}_d(q)$ (i.e. the system as well as its generator are proper) and the leading coefficient (the coefficient of the highest power of q) is invertible.

Stability. An IO relationship generated by $S_0 = \{(u_0, y_0) \mid [\tilde{a}(q)\tilde{e}(q) \vdots -\tilde{b}_d(q)]\}$ is stable in signal space if every solution y_0 to $\tilde{a}(q)\tilde{e}(q)y_0 = 0$ approaches 0 when the time k approaches infinity. It should be noted that in general, the stability in time-varying case cannot be tested from the pointwise roots of $\tilde{a}(q)\tilde{e}(q)$ calculated at every time instant. It can be shown that the poles of time variant systems are not related to these pointwise roots (O'Brien and Iglesias, 2001) and therefore conclusions on general stability cannot be drawn. For example, it is possible to show, that even all roots of discrete system are inside the unit circle at all time instants k, the overall system is unstable (O'Brien and Iglesias, 2001).

Observability. Observability can be studied from upper triangular form of internal IO-relation:

$$S_i = \left\{ (u,(y_1,y_2)) \left| \begin{bmatrix} \widetilde{A}_1(r) & \widetilde{A}_2(r) & \vdots & -\widetilde{B}_1(r) \\ 0 & \widetilde{A}_4(r) & \vdots & -\widetilde{B}_2(r) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ u \end{bmatrix} = 0 \right. \right\} \tag{77}$$

If $\widetilde{A}_1$ is invertible as mapping, then the composition is observable. If the system is causal, then it is always possible to take $\widetilde{A}_1(r) = I$. For the NCS it was obtained:

$$S_i = \left\{ (u,v,x,y) \left| \begin{bmatrix} 1 & 0 & 0 & \vdots & -d(r) \\ 0 & 1 & -\widetilde{e}(q) & \vdots & 0 \\ 0 & 0 & \widetilde{a}(q)\widetilde{e}(q) & \vdots & -\widetilde{b}_d(q) \end{bmatrix} \begin{bmatrix} v \\ x \\ y \\ u \end{bmatrix} = 0 \right. \right\} \tag{78}$$

From this obviously $\widetilde{A}(r) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and therefore, the internal IO-relation is observable.

Controllability. It can be defined, that the relation generated by $[A(r) \;\vdots\; -B(r)]$ is controllable, if $A(r)$ and $B(r)$ are left coprime, i.e their common left divisors are all unimodular. If $L(r)$ is the greatest common left divisor of $A(r)$ and $B(r)$ and $[A(r) \;\vdots\; -B(r)]$ is factorized as $[A(r) \;\vdots\; -B(r)] = L(r)[A_1(r) \;\vdots\; -B_1(r)]$, then $A_1(r), B_1(r)$ are left coprime. Now if $L(r)$ is unimodular, then the relation is controllable. As was mentioned, the skew polynomial matrix is unimodular if it is invertible as a skew polynomial matrix. Furthermore, the skew polynomial matrix is invertible if its diagonal values in triangular form are invertible.

6.2.5 Observer design for NCS

In this section, an observer for the NCS will be constructed. Consider the composition of Figure 27 and suppose that only the overall input $u_0 = u$ and output

$y_0 = y$ are known. The observer design problem is to construct a dynamical system S_{obs}, an observer with two inputs y_0 and u_0 so that its output $\hat{x}$ estimates x i.e. the error is as small as possible and stable irrespective of the input u_0.

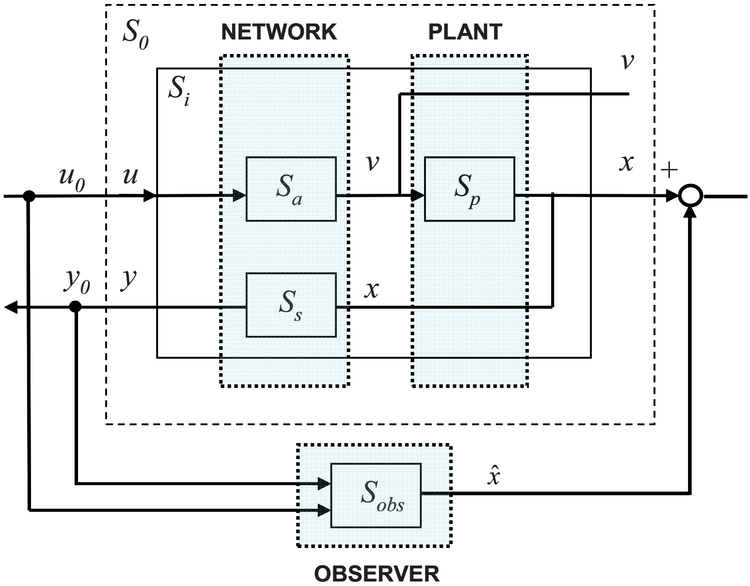

Figure 27. Observer design problem for a NCS.

There are many different solutions to the estimation problem. The observer type estimators are based on the system model so that the observer model and the system model belong to the same class of systems. It is natural to require that the correct $\hat{x}$ is a possible output of the observer.

Let us start from the following internal IO-relation S_i for the NCS in the upper triangular form:

$$S_i = \left\{ (u,(x,y)) \mid \begin{bmatrix} 1 & -\tilde{e}(q) & \vdots & 0 \\ 0 & \tilde{a}(q)\tilde{e}(q) & \vdots & -b_d(q) \end{bmatrix} \begin{bmatrix} x \\ y \\ u \end{bmatrix} = 0 \right\} \tag{79}$$

Let the observer S_{obs} to be designed be given by the IO-relation:

$$S_{obs} = \left\{ ((u_0, y_0), \hat{x}) \mid \begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \end{bmatrix} \begin{bmatrix} \hat{x} \\ y_0 \\ u_0 \end{bmatrix} = 0 \right\} \tag{80}$$

It has been shown by Ylinen (1980) that each generator $\begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \end{bmatrix}$ of the observer S_{obs} for the system given by (77) has to satisfy:

$$\begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \\ 0 & \vdots & \widetilde{A}_4(q) & -\widetilde{B}_2(q) \end{bmatrix} = \begin{bmatrix} T_1(q) & T_2(q) \\ 0 & I \end{bmatrix} \begin{bmatrix} \widetilde{A}_1(q) & \widetilde{A}_2(q) & \vdots & -\widetilde{B}_1(q) \\ 0 & \widetilde{A}_4(q) & \vdots & -\widetilde{B}_2(q) \end{bmatrix} \tag{81}$$

or in the NCS case

$$\begin{aligned} \begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \\ 0 & \vdots & \widetilde{a}(q)\widetilde{e}(q) & -\widetilde{b}_d(q) \end{bmatrix} &= \begin{bmatrix} T_1(q) & T_2(q) \\ 0 & I \end{bmatrix} \begin{bmatrix} 1 & -\widetilde{e}(q) & \vdots & 0 \\ 0 & \widetilde{a}(q)\widetilde{e}(q) & \vdots & -\widetilde{b}_d(q) \end{bmatrix} \\ &= \begin{bmatrix} I & T_2(q) \\ 0 & I \end{bmatrix} \begin{bmatrix} T_1(q) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} 1 & -\widetilde{e}(q) & \vdots & 0 \\ 0 & \widetilde{a}(q)\widetilde{e}(q) & \vdots & -\widetilde{b}_d(q) \end{bmatrix} \end{aligned} \tag{82}$$

where $T_1(q)$, and $T_2(q)$ are design matrices that can be changed in order to get desirable performance. The matrix $T_1(q)$ affects the stability of the estimation error and the matrix $T_2(q)$ is used to achieve a proper (causal, if *r* is used) observer. (The generator is proper if the degrees on the left side are not lower than the degree on the right hand side.) The overall aim is to achieve such behavior of the observer that it is robust with respect to parameter variations. Furthermore, it can be shown that the estimation error satisfies:

$$T_1(q)\widetilde{A}_1(q)(x-\hat{x}) = T_1(q)(x-\hat{x}) = 0\,. \tag{83}$$

Thus the design problem can be changed to the construction of the matrices $T_1(q)$ and $T_2(q)$. The design procedure can be as follows:

1. Choose a matrix $T_1(q)$ determining the error dynamics.
2. Use elementary row operations (multiplication by $\begin{bmatrix} I & T_2(q) \\ 0 & I \end{bmatrix}$) to realize the properness provided that the order of $T_1(q)\widetilde{A}(q) = T_1(q)$ is high enough.

3. If the properness cannot be achieved, the generator is multiplied by a new $T_1(q)$ and the use of elementary row operations is continued until the satisfactory result is obtained.

Illustrative example 1. Open loop observer.

In an open loop observer, the observer is driven only by the input u_0, no information of x is used and the dynamics of the observer are the same as the process dynamics. Therefore, the error dynamics of the observer are the same as of process model, or $T_1(q) = \tilde{a}(q)$.

A candidate for the observer is:

$$\begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{a}(q) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tilde{e}(q) & \vdots & 0 \\ 0 & \tilde{a}(q)\tilde{e}(q) & \vdots & -\tilde{b}_d(q) \end{bmatrix}$$
$$= \begin{bmatrix} \tilde{a}(q) & \vdots & -\tilde{a}(q)\tilde{e}(q) & 0 \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix} \tag{84}$$

So the generator for the observer is:

$$\begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \end{bmatrix} = \begin{bmatrix} \tilde{a}(q) & \vdots & -\tilde{a}(q)\tilde{e}(q) & 0 \end{bmatrix} \tag{85}$$

or

$$\tilde{a}(q)\hat{x} = -\tilde{a}(q)\tilde{e}(q)x + \tilde{b}(q)u_0 \tag{86}$$

The degree of the right hand side is higher than on left hand side. The observer is not proper. The degree can be reduced by adding the second row to the first row (or multiplying by $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $T_2(q) = 1$):

$$\begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{a}(q) & -\tilde{a}(q)\tilde{e}(q) & \vdots & 0 \\ 0 & \tilde{a}(q)\tilde{e}(q) & \vdots & -\tilde{b}_d(q) \end{bmatrix}$$
$$\begin{bmatrix} \tilde{a}(q) & \vdots & 0 & -\tilde{b}_d(q) \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix} \tag{87}$$

The obtained open loop observer is now proper and can be implemented as:

$$\tilde{a}(q)\hat{x} = b_d(q)u_0 \tag{88}$$

Illustrative example 2. Smith predictor.

In the Smith predictor, the observer is driven by the input u_0, and the value of x is also fed to the predictor and the dynamics of the observer are the same as the process dynamics. The error dynamics of the observer are the same as of the overall IO-relation, or

$$T_1(q) = \tilde{e}(q)\tilde{a}(q)\,. \tag{89}$$

A candidate for the observer is:

$$\begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{e}(q)\tilde{a}(q) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tilde{e}(q) & \vdots & 0 \\ 0 & \tilde{a}(q)\tilde{e}(q) & \vdots & -\tilde{b}_d(q) \end{bmatrix}$$
$$= \begin{bmatrix} \tilde{e}(q)\tilde{a}(q) & \vdots & -\tilde{e}(q)\tilde{a}(q)\tilde{e}(q) & 0 \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix} \tag{90}$$

Again, the degree of the right hand side of observer is higher than on left hand side. The observer is not proper. The degree can be reduced by adding the second row multiplied by (-1) to the first row, and then once more by multiplying the second row by $\tilde{e}(q)$ from the left and adding to the first row (or multiplying by

$$\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \tilde{e}(q) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \tilde{e}(q)-1 \\ 0 & 1 \end{bmatrix},\ T_2(q) = \tilde{e}(q)-1\,) \tag{91}$$

$$\begin{bmatrix} C(q) & \vdots & -D_1(q) & -D_2(q) \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix} = \begin{bmatrix} 1 & \tilde{e}(q)-1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{e}(q)\tilde{a}(q) & \vdots & -\tilde{e}(q)\tilde{a}(q)\tilde{e}(q) & 0 \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix}$$
$$= \begin{bmatrix} \tilde{e}(q)\tilde{a}(q) & \vdots & -\tilde{a}(q)\tilde{e}(q) & -(\tilde{e}(q)-1)\tilde{b}_d(q) \\ 0 & \vdots & \tilde{a}(q)\tilde{e}(q) & -\tilde{b}_d(q) \end{bmatrix} \tag{92}$$

The obtained Smith predictor is now proper and can be implemented as:

$$\tilde{e}(q)\tilde{a}(q)\hat{x} = \tilde{a}(q)\tilde{e}(q)y_0 + (\tilde{e}(q)-1)b_d(q)u_0 \tag{93}$$

6.2.6 Feedback controller design

In this section, the feedback controller is designed. It is assumed that the time invariant controller has been designed, and the goal is to make the controller time variant. The obtained feedback controller uses the estimate given by the observer designed in the previous section. According to the separation theorem, the feedback and observer can be designed independently (Ylinen, 2000). This also holds for the time-varying case.

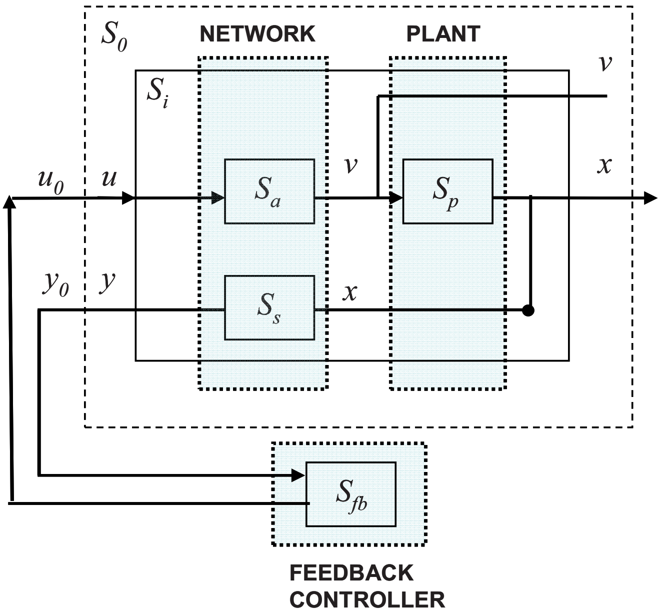

Figure 28. Feedback composition, observers output as input.

Next, consider the feedback controller design for the overall IO-relation S_o:

$$S_o = \left\{ (u_0, y_0) \,|\, \left[\widetilde{a}(q)\widetilde{e}(q) \vdots - \widetilde{b}_d(q) \right] \begin{bmatrix} y_0 \\ u_0 \end{bmatrix} = 0 \right\} \tag{94}$$

The problem is to construct a relation S_{fb}, a feedback controller, with input y_0 and output u_0 so that the overall system behaves satisfactorily. The feedback composition is depicted by Figure 28. The feedback controller is assumed again to belong to the same class of relations than the controlled relation S_o.

It can be shown that the generator $[C(q) \;\vdots\; -D(q)]$ of the feedback controller satisfies:

$$\underbrace{\begin{bmatrix} A_1(q) & \vdots & -B_1(q) \\ -D(q) & \vdots & C(q) \end{bmatrix}}_{A(q)} = \underbrace{\begin{bmatrix} I & 0 \\ T_3(q) & T_4(q) \end{bmatrix}}_{T(q)} \underbrace{\begin{bmatrix} A_1(q) & -B_1(q) \\ X(q) & Y(q) \end{bmatrix}}_{Q(q)}$$
$$= \begin{bmatrix} I & 0 \\ T_3(q) & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & T_4(q) \end{bmatrix} \begin{bmatrix} A_1(q) & -B_1(q) \\ X(q) & Y(q) \end{bmatrix} \tag{95}$$

for some $[T_3(q) \quad T_4(q)]$ and a unimodular $Q(q)$. In (95) $[A_1(q) \quad A_2(q)]$ is a generator for the controllable subsystem of S_o, i.e. $A_1(q)$ and $A_2(q)$ are left coprime. Analogously to the observer design, also here $T_3(q)$ and $T_4(q)$ are design matrices that can be changed in order to get desirable performance. The matrix $T_4(q)$ affects the closed loop behavior (together with uncontrollable part $L(q)$), and matrix $T_3(q)$ is used to achieve a proper (causal if r is used) feedback controller. In order to guarantee the robustness, the controller should be at least proper but preferably strictly proper. This can be achieved by choosing T_4 of high order and using T_3 for decreasing the degrees.

Assume that the feedback controller has been designed for time invariant part of the system S_p:

$$S_p = \left\{ (v,x) \mid [a(r) \vdots -b(r)] \begin{bmatrix} x \\ v \end{bmatrix} = 0 \right\} \tag{96}$$

The obtained controller $[c_{Pfb}(r) \vdots -d_{Pfb}(r)]$ is in the form of:

$$\begin{bmatrix} a(r) & -b(r) \\ -d_{Pfb}(r) & c_{Pfb}(r) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ t_3(r) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & t_4(r) \end{bmatrix} \begin{bmatrix} a(r) & -b(r) \\ x_{Pfb}(r) & y_{Pfb}(r) \end{bmatrix} \tag{97}$$

This can be used as a starting point for time variant controller design, and designing the feedback controller $[c(r) \vdots -d(r)]$ from:

$$\begin{bmatrix} a(r) & -b(r)d(r) \\ -d(r) & c(r) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ t_3(r) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & t_4(r) \end{bmatrix} \begin{bmatrix} a(r) & -b(r)d(r) \\ x_{Pfb}(r) & y_{Pfb}d(r) \end{bmatrix} \tag{98}$$

Note that all stabilizing feedback controllers for the networked system can be generated by:

$$\begin{bmatrix} c(r) & \vdots & -d(r) \end{bmatrix} = \begin{bmatrix} -t_3(r)b(r)d(r) + t_4(r)y_{Pfb}(r)d(r) & \vdots & t_3(r)a(r) + t_4 x_{Pfb}(r) \end{bmatrix} \tag{99}$$

varying the parameters $t_3(r)$, and $t_4(r)$.

6.3 Simulations of the polynomial system-based approach in the Matlab environment

Simulations are presented in this section to illustrate the implementation of the time variant controller and observer designs. First, an open loop observer is constructed and its performance illustrated with an example. Next, a Smith predictor is constructed for the first and second order systems in order to illustrate the complexity. After this, a time variant feedback compensator design is performed and combined with the observer design.

6.3.1 Transmission delay simulations

One of the assumptions concerning the behavior of the transmission delay was that each value $u(k)$ should appear only once in the values of $v(k)$. This means that no values can arrive simultaneously to the process, nor there can be any vacant sampling instant where no information has been received. This assumption had to be made in order to obtain a mathematical tool to handle time-variant behavior.

In order to illustrate this assumption, consider the lower left part of Figure 29 where a sinusoidal input signal u is shown that is sent over the network. After the signal has been transmitted through the network, the delay of which behaves according to the upper part of Figure 29, the received signal is shown in the right part of the figure. In the simulation, the delay was set to vary between 1 and 3 samples; with no emphasis being paid on the sequence of variation. As a result, it is possible that at some time instants, several values from the input signal arrive simultaneously; on the other hand, there are some vacant time instants at which no values are received. Therefore, the transmission delay is not invertible.

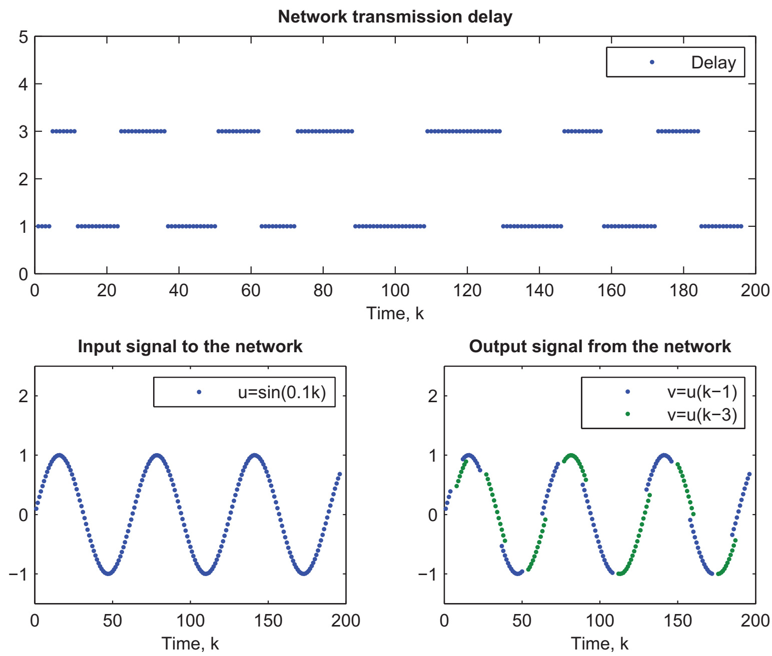

Figure 29. Simulation of a non-invertible transmission delay.

Figure 30 illustrates the simulation of an invertible transmission delay. In the simulation, the delay was set to vary from 1 to 9 sample instants such that in the output from the network each value appears only once. A sinusoidal input signal *u* that is sent over the network is illustrated in the lower left part of Figure 30. After the signal has been transmitted through the network, the delay of which behaves according to the upper part of the Figure 30, the received signal is shown in the lower right part of Figure 30. From the figure, it can be concluded that even though the shape of the signal remains the same, there is considerably more distortion.

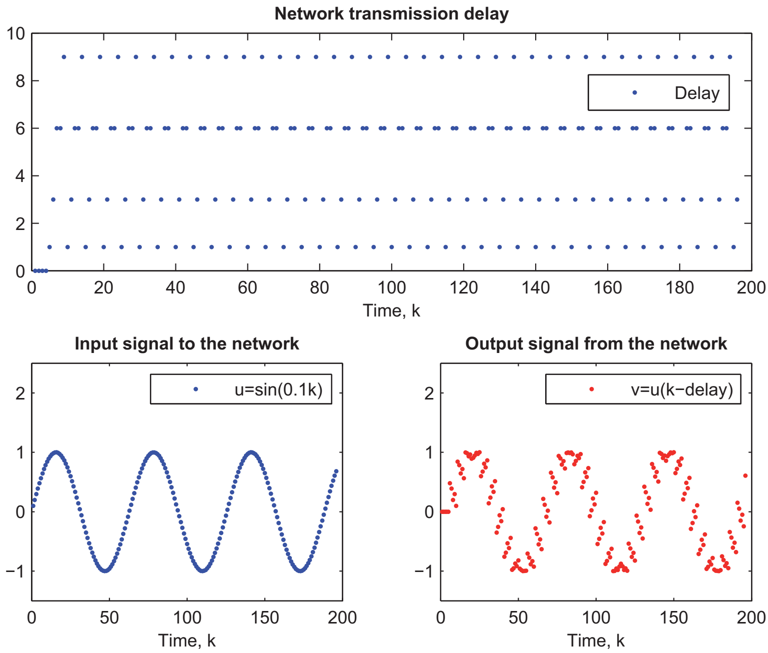

Figure 30. Simulation with an invertible transmission delay.

The delay sequence of Figure 30 that is repeated continuously is illustrated in Table 5. The second column of the table shows the sending order at which the information is sent through the network. The third column illustrates the delay value for the sent signal and the fourth column shows the receive order, respectively. According to the table, the first value is delayed for nine time instants; the second and third values are delayed for six time instants; the fourth for three time instants; and finally the fifth input value is delayed for one time instant. As can be seen from the third column, all values are eventually received but in a different order and the input set is also shifted as a whole by five time instants.

Table 5. Delay variation to achieve an invertible output signal.

#	Sending order	Delay	Receive order
1	$u(k-4)$	9	$v(k+1)=u(k)$
2	$u(k-3)$	6	$v(k+2)=u(k-1)$
3	$u(k-2)$	6	$v(k+3)=u(k-3)$
4	$u(k-1)$	3	$v(k+4)=u(k-2)$
5	$u(k)$	1	$v(k+5)=u(k-4)$

6.3.2 Open loop observer

In this section the performance of the time-variant open loop observer is illustrated with a numerical example. Consider the time invariant system described by:

$$(1-0.80r)x = 0.20rv \tag{100}$$

where $a(r)=(1-0.80r)$, $b(r)=0.20r$, and $\tilde{a}(q)=(q-0.80)$, $\tilde{b}(q)=0.20$.

From (60), the time-varying controller-to-plant transmission delay is:

$$v(k)=u(k-\theta(k))=(r^{\theta(.)}u)(k) \text{ or } v=(d_0+d_1r+\ldots+d_nr^n)u=d(r)u \tag{101}$$

where the coefficients are zero otherwise but $d_{\theta(k)}(k)=1$

Then the open loop observer is given by:

$$a(r)\hat{x}=b(r)d(r)u_0 \tag{102}$$

or

$$\begin{aligned}(1-0.80r)\hat{x}(k)&=0.20rr^{\theta(.)}u_0(k)=(0.20r^{\theta(.-1)+1})u_0(k)\\ \hat{x}(k)-0.80\hat{x}(k-1)&=0.20u_0(k-1-\theta(k-1))\end{aligned} \tag{103}$$

The simulation result for the obtained open loop observer is presented in Figure 31. The upper part of the figure shows the network transmission delay variation; the lower part of the figure shows the process and observer responses to the step input change that is sent through the network. The network was simulated in such a way that each input value to the network appears only once in the output. From the process response it can be seen that the response is distorted around the time instant of input change (from 0 to 1 and vice versa); due to the network delay the input changes from 0 to 1. From the response of the open loop observer it can be seen that the observer is able to predict the state of the process. The dynamics of the prediction error closely follow the dynamics of the process.

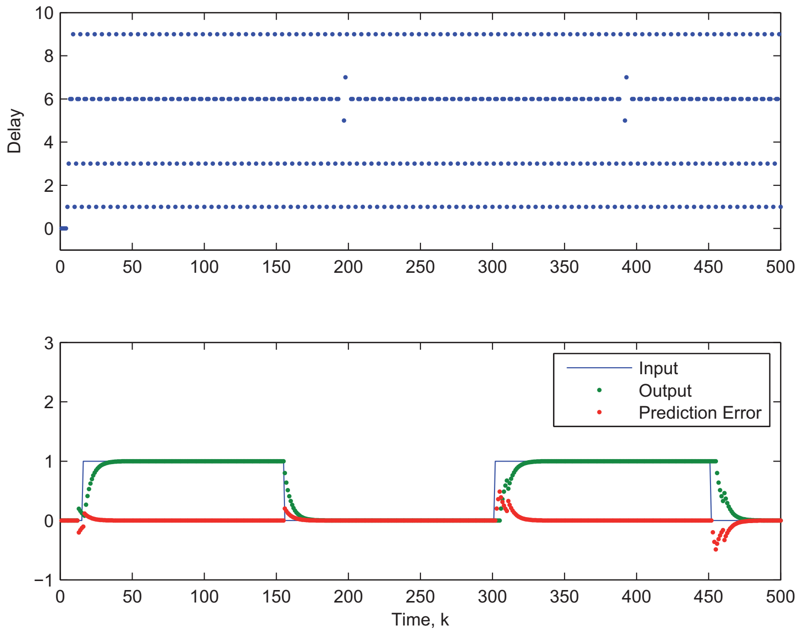

Figure 31. Input delay and output prediction and the performance of the observer.

6.3.3 Smith predictor

This section presents a simulation result for the Smith predictor for the first order system controlled over the network that is designed according to the procedures in Section 6.2.5. Consider first an order process given by

$$(1-0.99r)x = 0.01rv \tag{104}$$

from this

$$\begin{aligned} a(r) &= (1-0.99r) \qquad & b(r) &= 0.01r \\ \tilde{a}(q) &= (q-0.99) \qquad & \tilde{b}(q) &= 0.01 \end{aligned} \tag{105}$$

The Smith predictor is given by (93)

$$\tilde{e}(q)\tilde{a}(q)\hat{x} = \tilde{a}(q)\tilde{e}(q)y_0 + (\tilde{e}(q)-1)b_d(q)u_0 \tag{106}$$

where $\widetilde{a}(q)$ is of first order, therefore $b_d(q) = qb(r)d(r)$. The transmission delays are $\widetilde{e}(q) = q^{v_2(.)}$, $d(r) = r^{\theta_1(.)}$.

Inserting the above to the predictor equation:

$$(q^{v_2(.)}(q-0.99))\hat{x}(k) = ((q-0.99)q^{v_2(.)})y_0(k) + ((q^{v_2(.)}-1)\cdot q\cdot 0.01rr^{\theta_1(.)})u_0$$

Multiplying by $rr^{\theta_2(.)}$ from the left to get the predictor for implementation:

$$\begin{aligned} & rr^{\theta_2(.)}(q^{v_2(.)}(q-0.99))\hat{x}(k) = rr^{\theta_2(.)}((q-0.99)q^{v_2(.)})y_0(k) \\ & + rr^{\theta_2(.)}((q^{v_2(.)}-1)\cdot q\cdot 0.01rr^{\theta_1(.)})u_0(k) \end{aligned} \tag{107}$$

Rearranging by taking into account $q^{v_i(.)}r^{\theta_i(.)} = r^{\theta_i(.)}q^{v_i(.)} = 1$, and $q^i r^i = r^i q^i = 1$ gives:

$$\hat{x}(k) - 0.99r\hat{x}(k) = rr^{\theta_2(.)}qq^{v_2(.)}y_0(k) - 0.99ry_0(k) + 0.01rr^{\theta_1(.)}u_0(k) - 0.01rr^{\theta_2(.)}r^{\theta_1(.)}u_0(k) \tag{108}$$

After using the skew polynomial non-commutativity property $rb = r_K(b) + 0_K(b)$ repeatedly we obtain:

$$\begin{aligned} & \hat{x}(k) - 0.99\hat{x}(k-1) = y_0(k-\theta_2(k-1)+v_2(k-\theta_2(k-1))) - 0.99y_0(k-1) \\ & + 0.01u_0(k-1-\theta_1(k-1)) - 0.01u_0(k-1-\theta_2(k-1)-\theta_1(k-1-\theta_2(k-1))) \end{aligned} \tag{109}$$

And for implementation the following equation is obtained:

$$\begin{aligned} & \hat{x}(k) = 0.99\hat{x}(k-1) + y_0(k-\theta_2(k-1)+v_2(k-\theta_2(k-1))) - 0.99y_0(k-1) \\ & + 0.01u_0(k-1-\theta_1(k-1)) - 0.01u_0(k-1-\theta_2(k-1)-\theta_1(k-1-\theta_2(k-1))) \end{aligned} \tag{110}$$

Example 2. Consider a more complicated example and a second order system described with

$$(1-0.83r+0.0055r^2)x = (0.15r+0.033r^2)v \tag{111}$$

$$a(r) = 1-0.83r+0.0055r^2, \quad b(r) = 0.15r+0.033r^2, \quad \widetilde{a}(q) = q^2-0.83q+0.0055,$$

$$b(r) = 0.15q^2+0.033q \tag{112}$$

Again, the Smith predictor is given by:

$$\widetilde{e}(q)\widetilde{a}(q)\hat{x} = \widetilde{a}(q)\widetilde{e}(q)y_0 + (\widetilde{e}(q)-1)b_d(q)u_0 \tag{113}$$

where $\widetilde{e}(q) = q^{v_2(.)}$ and $d(r) = r^{\theta_1(.)}$. $\widetilde{a}(q)$ is of second order, therefore $b_d(q) = q^2 b(r)d(r)$. Inserting these to the predictor equation gives:

$$\begin{aligned}&(q^{v_2(.)}(q^2 - 0.83q + 0.0055))\hat{x}(k) = ((q^2 - 0.83q + 0.0055)q^{v_2(.)})y_0(k) + ((q^{v_2(.)}\\&-1)\cdot q^2 \cdot (0.15r + 0.033r^2)r^{\theta_1(.)})u_0(k)\end{aligned} \tag{114}$$

Multiplying by $r^2 r^{\theta_2(.)}$ from the left to obtain the formula for implementation:

$$\begin{aligned}&r^2 r^{\theta_2(.)}(q^{v_2(.)}(q^2 - 0.83q + 0.0055))\hat{x}(k) = r^2 r^{\theta_2(.)}((q^2 - 0.83q + 0.0055)q^{v_2(.)})y_0(k)\\&+ r^2 r^{\theta_2(.)}((q^{v_2(.)} - 1)\cdot q^2 \cdot (0.15r + 0.033r^2)r^{\theta_1(.)})u_0(k)\end{aligned} \tag{115}$$

Rearranging and taking into account $q^{v_i(.)} r^{\theta_i(.)} = r^{\theta_i(.)} q^{v_i(.)} = 1$, $q^i r^i = r^i q^i = 1$, gives:

$$\begin{aligned}&r^2(q^2 - 0.83q + 0.0055)\hat{x}(k) = r^2 r^{\theta_2(.)} q^2 q^{v_2(.)} y_0(k) - 0.83 r^2 r^{\theta_2(.)} q q^{v_2(.)} y_0(k)\\&+ 0.0055 r^2 r^{\theta_2(.)} q^{v_2(.)} y_0(k) + r^2 r^{\theta_2(.)} q^{v_2(.)} q^2 \cdot 0.15 r r^{\theta_1(.)} u_0(k) - r^2 r^{\theta_2(.)} q^2 \cdot 0.15 r r^{\theta_1(.)} u_0(k)\\&+ r^2 r^{\theta_2(.)} q^{v_2(.)} q^2 \cdot 0.033 r^2 r^{\theta_1(.)} u_0(k) - r^2 r^{\theta_2(.)} q^2 \cdot 0.033 r^2 r^{\theta_1(.)} u_0(k)\end{aligned} \tag{116}$$

$$\begin{aligned}&\hat{x}(k) - 0.83 r\hat{x}(k) + 0.0055 r^2 \hat{x}(k) = r^2 r^{\theta_2(.)} q^2 q^{v_2(.)} y_0(k) - 0.83 r^2 r^{\theta_2(.)} q q^{v_2(.)} y_0(k)\\&+ 0.0055 r^2 y_0(k) + r^2 q^2 \cdot 0.15 r r^{\theta_1(.)} u_0(k) - 0.15 r^2 r^{\theta_2(.)} q^2 \cdot r r^{\theta_1(.)} u_0(k)\cdot\\&+ 0.033 r^2 q^2 \cdot r^2 r^{\theta_1(.)} u_0(k) - 0.033 r^2 r^{\theta_2(.)} q^2 \cdot r^2 r^{\theta_1(.)} u_0(k)\end{aligned} \tag{117}$$

Rearranging and taking into account the skew polynomial product property $rb = r_K(b)r + 0_K$ gives:

$$\begin{aligned}&\hat{x}(k) - 0.83\hat{x}(k-1) + 0.0055\hat{x}(k-2) = y_0(k - \theta_2(k-2) + v_2(k - \theta_2(k-2)))\\&- 0.83 y_0(k - 1 - \theta_2(k-2) + v_2(k - 1 - \theta_2(k-2))) + 0.0055 y_0(k-2) + 0.15\cdot u_0(k - 1 - \theta_1(k-1))\\&- 0.15 u_0(k - 1 - \theta_2(k-2) - \theta_1(k - 1 - \theta_2(k-2))) + 0.033 u_0(k - 2 - \theta_1(k-2)) - 0.033 u_0(k-2\\&- \theta_2(k-2) - \theta_1(k - 2 - \theta_2(k-2)))\end{aligned} \tag{118}$$

Predictor for implementation:

$$
\begin{aligned}
\hat{x}(k) &= 0.83\hat{x}(k-1) - 0.0055\hat{x}(k-2) + y_0(k-\theta_2(k-2)+v_2(k-\theta_2(k-2))) \\
&- 0.83y_0(k-1-\theta_2(k-2)+v_2(k-1-\theta_2(k-2))) + 0.0055y_0(k-2) + 0.15\cdot u_0(k-1-\theta_1(k-1)) \\
&- 0.15u_0(k-1-\theta_2(k-2)-\theta_1(k-1-\theta_2(k-2))) + 0.033u_0(k-2-\theta_1(k-2)) - 0.033u_0(k-2 \\
&- \theta_2(k-2)-\theta_1(k-2-\theta_2(k-2)))
\end{aligned}
\tag{119}
$$

The step response of the PI controlled second order system with this Smith predictor is illustrated in Figure 32. The first row of the figure illustrates the simulated delay variation in the sensor and actuator paths while the second row illustrates the performance of the PI controlled system when no predictor has been implemented. From the figure it can concluded that the nominal controller is able to maintain the performance of the system; however, the network affects the performance and the oscillation is relatively high. Row 3 illustrates the performance when the Smith predictor scheme has been turned on with the same nominal PI controller. From the figure it can be directly seen that performance improves significantly and the effect of the network is not observable in the process response. Finally, in row 4 the step response of the process controller with the Smith predictor and where the gain of the PI controller was doubled. From the response it can be concluded that the response is faster and that the performance of the system is still unaffected by the network.

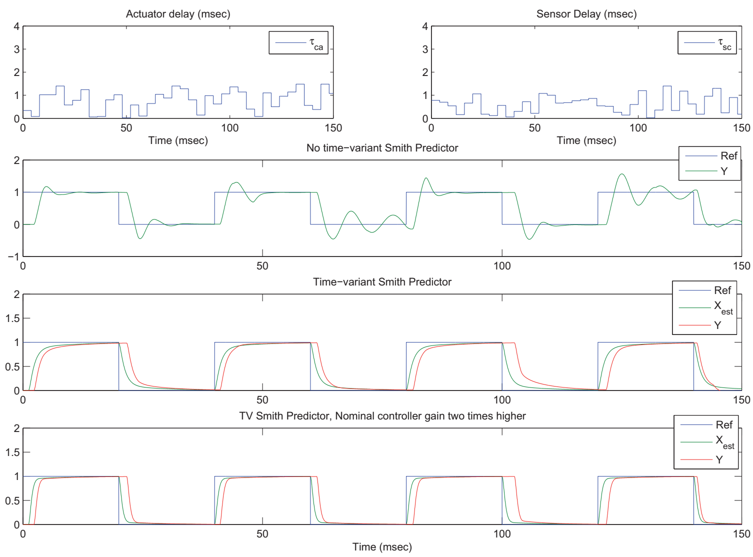

Figure 32. Second order system, basic simulations.

Similarly as presented in Section 5.3.1, a study was performed to evaluate the influence of the delay measurement error on the control performance of the Smith predictor scheme. Consider Figure 33 where four simulation results are illustrated increasing the measurement error gradually from 15% to 55% of its real value. The first row illustrates the performance when the measurement error was 15%; the second row is with the error of 25%; the third 35%; and in the fourth the error is 55%. Four graphs are shown on each row: the first two illustrate the actual and measured delays in the sensor and actuator paths; the third graph shows the step response of the process and the Smith predictor state estimate; and finally the fourth graph shows the prediction error evolution.

From the figure it can be concluded that as the error in the delay measurement increases, the performance of the Smith predictor based approaches is significantly reduced and becomes very oscillative at the point with the measurement error of 55%. The dynamics of the prediction error are still of the second order, but with significantly more oscillation.

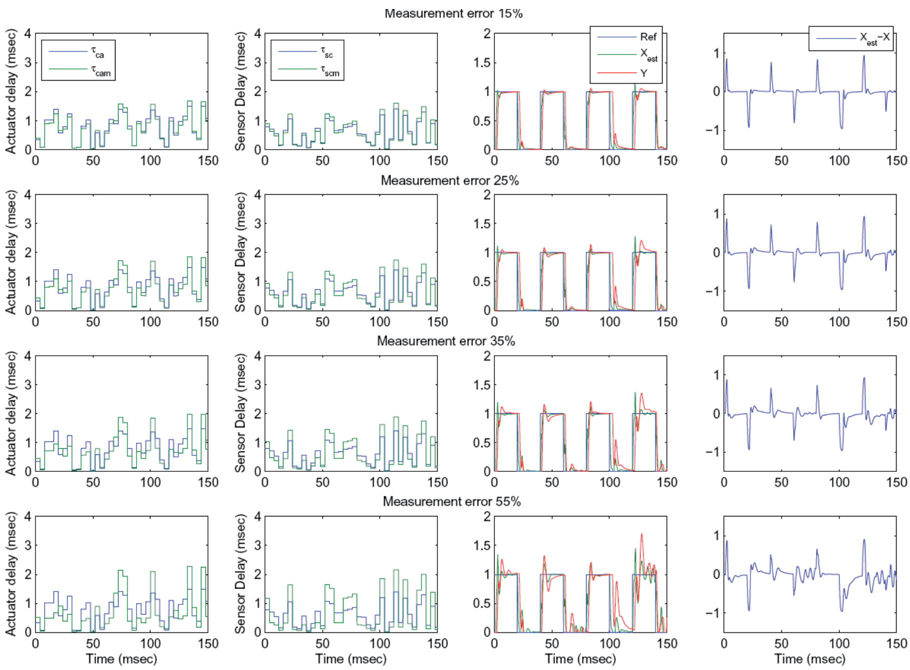

Figure 33. Second order system, measurement error simulations.

6.3.4 Feedback compensator design

Consider the following system

$$
\begin{aligned}
a(q) &= (q - 0.99) \\
b(q) &= 0.01
\end{aligned}
\tag{120}
$$

The system is controlled over the network, where the information of the actuator path delay is known and varying according to $q^{\theta(.)}$. The goal is to design a time variant controller with polynomial systems theory, or

$$
\begin{bmatrix} \tilde{c}(q)a(q) & -b(q) \\ -d(q) & c(q) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ t_3(q) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & t_4(q) \end{bmatrix} \begin{bmatrix} \tilde{c}(q)a(q) & -b(q) \\ x_{Pfb}(q) & y_{Pfb}d(q) \end{bmatrix}
\tag{121}
$$

where $\tilde{c}(q) = q^{\theta(.)}$ and one of coefficients $\tilde{c}_0, \tilde{c}_1, \ldots \tilde{c}_n$ is one and all others zero, at k. The first step is to find matrix $P(q)$. This can be done with the following elementary column operations:

$$
\begin{aligned}
&\begin{bmatrix} q^{\theta(.)}q - 0.99q^{\theta(.)} & -0.01 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \xrightarrow{T_{12}(-99q^{\theta(.)})} \begin{bmatrix} q^{\theta(.)}q & -0.01 \\ 1 & 0 \\ -99q^{\theta(.)} & 1 \end{bmatrix} \\
&\xrightarrow{T_{12}(100q^{\theta(.)}q)} \begin{bmatrix} 0 & -0.01 \\ 1 & 0 \\ 100q^{\theta(.)}q - 99q^{\theta(.)} & 1 \end{bmatrix} \xrightarrow{\left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]} \begin{bmatrix} -0.01 & 0 \\ 0 & 1 \\ 1 & 100q^{\theta(.)}q - 99q^{\theta(.)} \end{bmatrix}
\end{aligned}
\tag{122}
$$

or

$$
P(q) = \begin{bmatrix} 0 & 1 \\ 1 & 100q^{\theta(.)}q - 99q^{\theta(.)} \end{bmatrix}
\tag{123}
$$

The next step is to find matrix $Q(q) = P(q)^{-1}$.

$$Q(q)=P(q)^{-1}=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}^{-1} T_{12}(100q^{\theta(.)}q)^{-1}\cdot T_{12}(-99q^{\theta(.)})^{-1}=$$
$$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}^{-1}\begin{bmatrix}1 & 0\\ 100q^{\theta(.)}q & 1\end{bmatrix}^{-1}\begin{bmatrix}1 & 0\\ -99q^{\theta(.)} & 1\end{bmatrix}^{-1}=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & -100q^{\theta(.)}q\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 99q^{\theta(.)}\\ 0 & 1\end{bmatrix}=$$
$$\begin{bmatrix}-100q^{\theta(.)}q+99q^{\theta(.)} & 1\\ 1 & 0\end{bmatrix} \tag{124}$$

Lets place root at $q^{\theta(.)}q=0.8$:

$$\begin{bmatrix}\tilde{c}(q)a(q) & -b(q)\\ -d(q) & c(q)\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & q^{\theta(.)}q-0.8\end{bmatrix}\begin{bmatrix}-100q^{\theta(.)}q+99q^{\theta(.)} & 1\\ 1 & 0\end{bmatrix}=\begin{bmatrix}-100q^{\theta(.)}q+99q^{\theta(.)} & 1\\ q^{\theta(.)}q-0.8 & 0\end{bmatrix} \tag{125}$$

Drop order by one:

$$\begin{bmatrix}\tilde{c}(q)a(q) & -b(q)\\ -d(q) & c(q)\end{bmatrix}=\begin{bmatrix}1 & 0\\ 1/100 & 1\end{bmatrix}\begin{bmatrix}-100q^{\theta(.)}q+99q^{\theta(.)} & 1\\ q^{\theta(.)}q-0.8 & 0\end{bmatrix}=\begin{bmatrix}-100q^{\theta(.)}q+99q^{\theta(.)} & 1\\ 99/100q^{\theta(.)}-0.8 & 1/100\end{bmatrix} \tag{126}$$

The controller is not proper, another root is needed. Let us place a root at $q=0.8$:

$$\begin{aligned}\begin{bmatrix}\tilde{c}(q)a(q) & -b(q)\\ -d(q) & c(q)\end{bmatrix}&=\begin{bmatrix}1 & 0\\ 0 & q-0.8\end{bmatrix}\begin{bmatrix}-100q^{\theta(.)}q+99q^{\theta(.)} & 1\\ 99/100q^{\theta(.)}-0.8 & 1/100\end{bmatrix}\\ &=\begin{bmatrix}-100q^{\theta(.)}q+99q^{\theta(.)} & 1\\ 99/100qq^{\theta(.)}-0.8q-99/125q^{\theta(.)}+16/25 & 1/100q-1/125\end{bmatrix}\end{aligned} \tag{127}$$

The order has to be dropped, however, since $q^{\theta(.)}q\neq qq^{\theta(.)}$ the order cannot be drop directly by constant multiplication. Suppose that:

$$\begin{aligned}q^{\theta(.)}&=(\tilde{c}_0+\tilde{c}_1q)\\ q^{\theta(.)}q&=(\tilde{c}_0+\tilde{c}_1q)q\\ qq^{\theta(.)}&=(\tilde{c}_0{}^q+\tilde{c}_1^q q)q\end{aligned} \tag{128}$$

In the third form a predictive q operator has been calculated on the coefficients. Inserting these to an equation, we obtain:

$$\begin{bmatrix} v(q)a(q) & -b(q) \\ -d(q) & c(q) \end{bmatrix} =$$

$$\begin{bmatrix} -100(\widetilde{c}_0+\widetilde{c}_1 q)q+99(\widetilde{c}_0+\widetilde{c}_1 q) & 1 \\ 99/100(\widetilde{c}_0^{\,q}+\widetilde{c}_1^{\,q} q)q-0.8q-99/125(\widetilde{c}_0+\widetilde{c}_1 q)+16/25 & 1/100q-1/125 \end{bmatrix} \tag{129}$$

And by dropping the order the following expression is obtained:

$$\begin{bmatrix} v(q)a(q) & -b(q) \\ -d(q) & c(q) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{100\widetilde{c}_1}\frac{99}{100}\widetilde{c}_1^{\,q} & 1 \end{bmatrix} =$$

$$\begin{bmatrix} -100(\widetilde{c}_0+\widetilde{c}_1 q)q+99(\widetilde{c}_0+\widetilde{c}_1 q) & 1 \\ 99/100(\widetilde{c}_0^{\,q}+\widetilde{c}_1^{\,q} q)q-0.8q-99/125(\widetilde{c}_0+\widetilde{c}_1 q)+16/25 & 1/100q-1/125 \end{bmatrix} \tag{130}$$

$$\begin{bmatrix} -100(\widetilde{c}_0+\widetilde{c}_1 q)q+99(\widetilde{c}_0+\widetilde{c}_1 q) & 1 \\ -d(q) & c(q) \end{bmatrix}$$

where

$$-d(q)=\left(-\frac{99}{100\widetilde{c}_1}\widetilde{c}_1^{\,q}\widetilde{c}_0+\frac{9801}{10000}\widetilde{c}_1^{\,q}\widetilde{c}_1+99/100\widetilde{c}_0^{\,q}-0.8-99/125\widetilde{c}_1\right)q$$
$$+\left(\frac{9801}{10000\widetilde{c}_1}\widetilde{c}_1^{\,q}\widetilde{c}_0-99/125\widetilde{c}_0+16/25\right) \tag{131}$$

and

$$c(q)=\frac{1}{c_1}\frac{99}{10000}\widetilde{c}_1^{\,q}+1/100q-1/125$$

Therefore the controller when represented as r operator is:

$$c(r)u=d(r)y$$

$$\left(\frac{1}{\widetilde{c}_1}\frac{99}{10000}\widetilde{c}_1^{\,r}r+1/100-1/125r\right)u=\left(\left(\frac{99}{100\widetilde{c}_1}\widetilde{c}_1^{\,r}\widetilde{c}_0-\frac{9801}{10000}\widetilde{c}_1^{\,r}\widetilde{c}_1-99/100\widetilde{c}_0^{\,r}+0.8+99/125\widetilde{c}_1\right)\right.$$
$$\left.+\left(-\frac{9801}{10000\widetilde{c}_1}\widetilde{c}_1^{\,r}\widetilde{c}_0+99/125\widetilde{c}_0-16/25\right)r\right)y \tag{132}$$

$$u(k)=\left(\frac{4}{5}-\frac{1}{\widetilde{c}_1}\frac{99}{100}\widetilde{c}_1^{\,q}\right)u(k-1)+\left(\frac{99}{\widetilde{c}_1}\widetilde{c}_1^{\,q}\widetilde{c}_0-\frac{9801}{100}\widetilde{c}_1^{\,q}\widetilde{c}_1-99\widetilde{c}_0^{\,q}+80+\frac{396}{5}\widetilde{c}_1\right)y(k)$$
$$+\left(-\frac{9801}{100\widetilde{c}_1}\widetilde{c}_1^{\,q}\widetilde{c}_0+\frac{396}{5}\widetilde{c}_0-64\right)y(k-1) \tag{133}$$

And the controller for implementation:

$$
\begin{aligned}
\tilde{c}_1 u(k) = {} & \left(\frac{4}{5}\tilde{c}_1 - \frac{99}{100}\tilde{c}_1^{\,q}\right)u(k-1) + \left(99\tilde{c}_1^{\,q}\tilde{c}_0 - \frac{9801}{100}\tilde{c}_1\tilde{c}_1(q) - 99\tilde{c}_1\tilde{c}_0(q) + 80\tilde{c}_1 + \frac{396}{5}\tilde{c}_1^{\,2}\right)y(k) \\
& + \left(-\frac{9801}{100}\tilde{c}_1(q)\tilde{c}_0 + \frac{396}{5}\tilde{c}_1\tilde{c}_0 - 64\tilde{c}_1\right)y(k-1)
\end{aligned}
\tag{134}
$$

The simulation result for this system is presented in Figure 34. The upper part of the figure illustrates the step response of the feedback system, the lower part shows the network delay variation. In simulations the cases where c_1 equaled to zero in the left hand side, $c_1 = 1$ was used. From the figure is can be concluded, that time variant feedback controller manages to handle high time variance induced by the network.

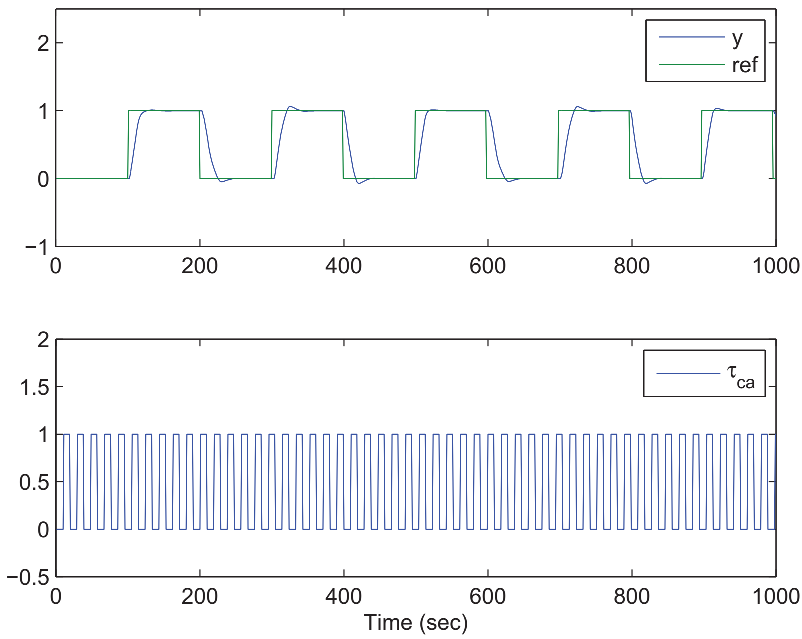

Figure 34. Response of feedback system with the time variant feedback controller.

Note that the obtained controller has no integrator. In order to take the integrator into account, the design has to be performed starting from the following system:

$$
\begin{aligned}
a(q) &= (q-0.99)(q-1) \\
b(q) &= 0.01
\end{aligned}
$$

7 Enhanced control of a NCS using delay estimations

The aim of this chapter is to propose a control approach that enables the performance improvement of a control application controlled over a switched Ethernet network by the utilization of an upper bound end-to-end delay estimate obtained from a network model. First, a methodology to obtain the upper bound of the network transmission delay in a switched Ethernet network is presented. Next, the obtained estimate is utilized in a robust controller design to achieve a controller that is tolerant for this upper bound delay. Finally, the performance of the proposed control scheme is illustrated with a Matlab simulation and with simulations on an experimental prototype.

7.1 Upper bound end-to-end delay estimation

The upper bound delay estimation algorithm applies ideas from the network calculus theory, see Cruz (1991), Boudec and Thiran (2001). The method has been developed in the PhD thesis of Jean-Philippe Georges and published in Georges, *et al.* (2005). The upper bound delay estimation method consists of two phases: in the first phase, the methodology for delay computation over a single switch is proposed; in the second phase, the upper bound delay over a switched Ethernet network consisting of several switches is calculated.

7.1.1 Traffic modeling and backlog for the Ethernet switch

The traffic arriving at the switch, both periodic and aperiodic, is modeled according to the "leaky bucket controller" principle. This means that data arrives at the leaky rate $R(t)$ only if the level of the bucket is less than the maximum bucket size σ, and leaves the bucket at a constant rate ρ (see

Figure **35**). The traffic is represented as arrival curves that give an estimate of the maximum amount of data that can arrive at the switch. It can be proved with the network calculus theory that if the traffic follows the leaky bucket principle and if the incoming rate of the traffic is limited by the port capacity C_{in}, these curves are affine and have the form:

$$b(t) = \min(C_{in}t, \quad \sigma + \rho t) \tag{135}$$

where σ is the maximum amount of data that can arrive in a burst to a component, ρ is an upper bound of the average rate of the traffic flow, and C_{in} is the capacity of a port. In the similar way, service curves are used to represent the minimum amount of data that can be processed at the component. Typical arrival and service curves are shown in Figure 36, where $b_1(t)$ and $b_2(t)$ are arrival curves of ports 1 and 2, and $b_{out}(t)$ is a service curve.

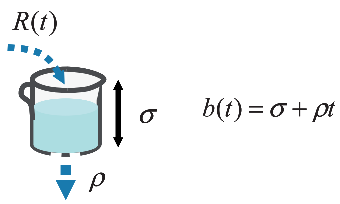

Figure 35. Traffic modeling according to the leaky bucket controller principle.

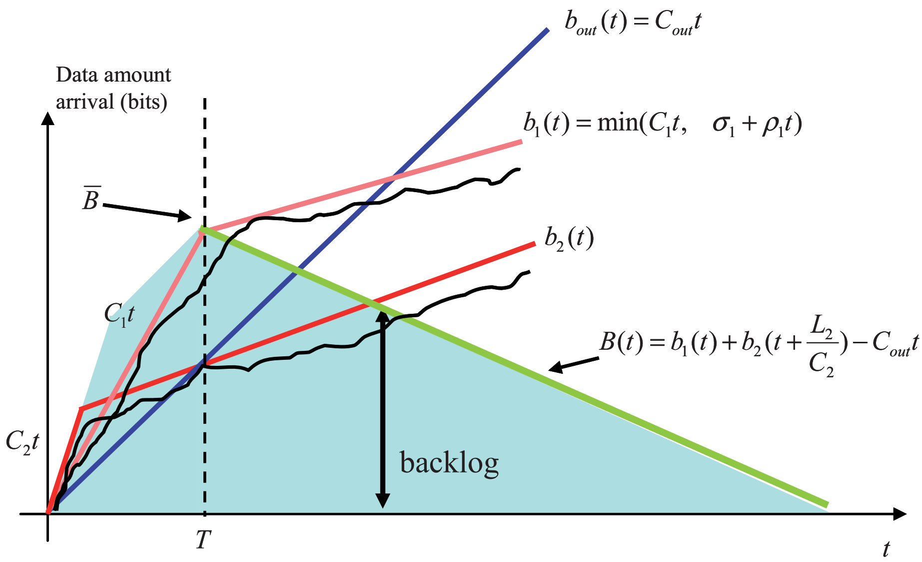

Figure 36. Arrival and service curves and backlog evolution inside the two-input FIFO multiplexer (Georges, *et al.*, 2005).

The approach to determine the upper bound over a switch is based on the evolution of a specific parameter, the backlog. The backlog is the number of bits waiting in the component, and it is a measure of congestion over the component. For the arrival curves in Figure 36, the backlog can be calculated by

$$B(t) = b_1(t) + b_2(t + L_2 / C_2) - C_{out}t \tag{136}$$

where $b_1(t)$ and $b_2(t)$ are the arrival curves of stream 1 and 2 at time t, L_2 is the maximum length of the frames, C_2 is the capacity of the import port 2, and C_{out} is the capacity of the output link. As an example, from Figure 36 it can be seen that the maximum backlog $\overline{B}$ corresponds to the time T when lines $C_1 t$ and $b_1(t)$ intercept:

$$C_1 T = \sigma_1 + \rho_1 T \rightarrow T = \frac{\sigma_1}{C_1 - \rho_1} \tag{137}$$

7.1.2 Maximum delay for crossing the Ethernet switch

Ethernet switches are modeled as a combination of basic components: multiplexers, demultiplexers and First In First Out (FIFO) queues, as shown in Figure 37. The first step in the switch modeling involves the determination of an upper bound delay for the crossing of each of the basic components. The upper bound delay over a switch is then the sum of the upper bound delays over the basic components:

$$\overline{D_{switch}} = \overline{D_{mux}} + \overline{D_{queue}} + \overline{D_{demux}} \tag{138}$$

where the notation $\overline{D}$ is used to represent the upper bound value of the delays over a component.

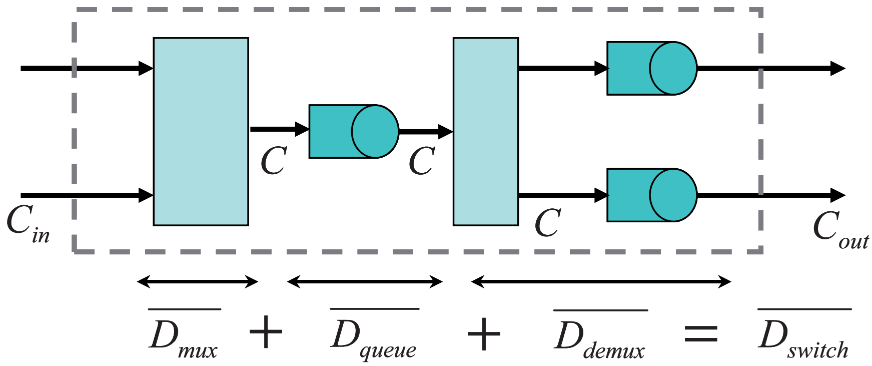

Figure 37. Model of a 2-port switch in a full duplex mode based on shared memory and a cut-through management (Georges, *et al.*, 2005).

Georges, *et al.* (2005) showed that in a FIFO m-inputs multiplexer, the delay for any incoming bit from the stream *i* is upper-bounded by:

$$\overline{D_{mux,i}} = \frac{1}{C_{out}} \min_k \overline{B_{mux,k}} \tag{139}$$

where $\overline{B_{mux,k}}$, $1 \le k \le m$, is an upper-bound of the backlog in the bursty period.

For $k = i$, the bursty period is defined by $T_i = \sigma_i / (C_i - \rho_i)$ and the backlog is upper-bounded by:

$$\overline{B}_{mux,i} = C_i T_i + \sum_{z=1;z\neq i}^{m} \left(b_z (T_i + \frac{L_z}{C_z}) \right) - C_{out} T_i = \sum_{z=1;z\neq i}^{m} \left(\sigma_z + \rho_z (T_i + \frac{L_z}{C_z}) \right) + (C_i - C_{out}) T_i =$$
$$\sum_{z=1;z\neq i}^{m} \left(\sigma_z + \rho_z (\frac{\sigma_i}{C_i - \rho_i} + \frac{L_z}{C_z}) \right) + (C_i - C_{out}) \frac{\sigma_i}{C_i - \rho_i} \tag{140}$$

where σ_z is the burstiness of the stream *z*, ρ_z is the average rate of arrival of the data of stream *z*, L is the maximum length of the frames of the stream, and C_i is the capacity of the import port *i*.

For $k \neq i$, $1 \leq k \leq m$, the bursty period is defined $T_k = \sigma_k /(C_k - \rho_k) - L_k / C_k$ and the backlog is upper-bounded by:

$$\begin{aligned}
\overline{B_{mux,k}} &= C_k T_k + \sum_{z=1;z\neq i}^{m} \left(b_z (T_k + \frac{L_z}{C_z}) \right) - C_{out} T_k - \rho_i \frac{L_i}{C_i} + L_k = \\
&= \sum_{z=1;z\neq k}^{m} \left(\sigma_z + \rho_z \left(T_k + \frac{L_z}{C_z} \right) \right) + (C_k - C_{out}) T_k - \rho_i \frac{L_i}{C_i} + L_k \\
&= \sum_{z=1;z\neq k}^{m} \left(\sigma_z + \rho_z \left(\frac{\sigma_k}{C_k - \rho_k} - \frac{L_k}{C_k} + \frac{L_z}{C_z} \right) \right) + (C_k - C_{out}) \left(\frac{\sigma_k}{C_k - \rho_k} - \frac{L_k}{C_k} \right) - \rho_i \frac{L_i}{C_i} + L_k
\end{aligned} \tag{141}$$

As can been seen from the equations, the upper bound is time independent since it only depends on the arrival curve parameters σ and ρ. The difference between the two expressions of the maximum backlog for the multiplexer comes from the worst case hypothesis that a frame of stream *i* has to wait for the forwarding of the frame to arrive simultaneously.

The formula for the upper bound delay for the FIFO queue can be derived directly from the multiplexer formula, as the FIFO queue can be considered as a multiplexer with just one input:

$$\overline{D_{queue}} = \overline{D_{mux}} = \frac{1}{C_{out}} \min\left(\overline{B}_{mux,i}\right) = \frac{1}{C_{out}} T \left(C_{in} - C_{out}\right) = \frac{1}{C_{out}} \frac{\left(C_{in} - C_{out}\right)}{C_{in} - \rho_{in}} \sigma_{in} \tag{142}$$

For the demultiplexer, it is assumed that the time required to route the output port is relatively negligible compared to the other delays, i.e. the demultiplexer does not generate delays.

7.1.3 Maximum end-to-end delays for crossing a switched Ethernet network

Computation of the upper bound end-to-end delays requires that special attention is paid to the input parameters of Equations (139-142). The maximum delay value $\overline{D}$ depends on the leaky bucket parameters: the maximum amount of traffic σ that can arrive in a burst, and the upper bound of the average rate of the traffic flow ρ. In order to calculate the maximum delay over the network, it is necessary that the envelope (σ, ρ) is known at every point in the network. However, as shown in Figure 38, only the initial arrival curve values (σ^0, ρ^0) are usually known, and the values for other arrival curves have to be determined. To calculate all the arrival curve values the following equations can be used:

$$\begin{aligned} \sigma_{out} &= \sigma_{in} + \rho_{in} D \\ \rho_{out} &= \rho_{in} \end{aligned} \tag{143}$$

For example, for the arrival curve (σ^1, ρ^1) in Figure 38 the envelope after the first switch is:

$$(\sigma^1, \rho^1) = (\sigma^0 + \rho^0 \overline{D}^1_{switch}, \rho^0) \tag{144}$$

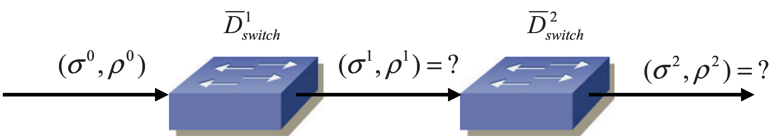

Figure 38. Burstiness along a switched Ethernet network (Georges, *et al.*, 2005).

The last part of the method used to obtain the upper-bounded delay estimate is the resolution of the burstiness characteristic of each flow at each point in the network. This will result in a linear equation system, from which the burstiness values can be solved:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{n2} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{145}$$

After the burstiness values are known, the upper bound end-to-end delays are obtained from

$$\overline{D}_i = \frac{\sigma_i^h - \sigma_i^o}{\rho_i} \tag{146}$$

where h is the number of crossed switches, and i represents a flow under consideration.

7.2 Robust control design for delay tolerance

The obtained end-to-end upper bound transmission delay will now be utilized in the control design. This control design method employed in this section is based on the H_∞ optimal control theory. One benefit of the proposed approach is that it will integrate both the optimal controller design and the delay tolerance into one framework. Also, the upper bound delay value obtained from the architectural investigation presented in the previous section can easily be integrated.

In this section it will be assumed that there is no uncertainty in the process model, nor process measurements, but only in the network transmission delay. Second assumption is that it is possible to lump together actuator and sensor path delays into a single uncertainty block by moving the sensor transmission delay block around the loop to the actuator delay path of the loop. This will greatly simplify controller synthesis and analysis. The third assumption is that only the SISO case is studied, which allows formulation of robust performance problem as mixed sensitivity problem. The mixed sensitivity problem then can be easily solved, and obtained controller is of relatively low order (compared to μ-synthesis). In the next chapter, where also the measurement errors will be taken into account all these assumptions will be relaxed.

7.2.1 General control configuration with uncertainty for networked control system

The system controlled by the networked controller, system G_{nw}, is considered to be a combination of the nominal plant (assumed to be fixed and certain) and uncertain (unknown, but bounded) effects of the network. As a nominal plant a SISO process is chosen. The network effect is represented as an unstructured dynamic uncertainty in a multiplicative form. This uncertainty representation form was chosen as it is well able to take into account errors due to missing dynamics at high frequencies. The other choice would be a parametric uncertainty form, which is used in case one parameter is in error and the order of the system is known. This, however, does not fit well for the delayed case as the delay makes the system of infinite dimensional.

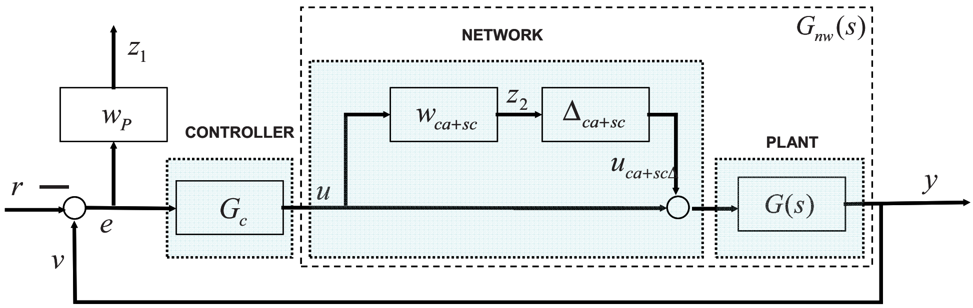

Figure 39. An approximated NCS for robust controller synthesis.

The obtained system is presented in Figure 39. In the figure, the transmission delays in the sensor and actuator paths have been lumped into one block by moving the sensor transmission delay block around the loop to the actuator delay path of the loop. In the SISO case, this does not change the input/output characteristics of the system, i.e. the H_∞ norm of the system will remain unchanged. The network block in the figure is a combination of weight function w_{ca+sc} and Δ_{ca+sc}, where $\Delta_{sc} = e^{-\bar{\tau}_{ca+sc}s} - 1$, $\|\Delta_{sc}\|_\infty < 1$. The perturbed plant becomes:

$$G_{nw} = e^{-(\tau_{ca}+\tau_{sc})s}G = (1+E)G = (1+w_{ca+sc}\Delta_{ca+sc})G \tag{147}$$

Therefore, the dynamic behavior of G_{nw} is described not only by a single linear time-invariant model but by with an infinite number of possible linear time-invariant plants models in a set Π. Set Π is norm-bounded and is generated by allowing H_∞ norm-bounded stable perturbations caused by the network to the nominal plant.

The generalized plant description P that will be used in the analysis and the H_∞ control synthesis for this system is:

$$\begin{bmatrix} z_2 \\ z_1 \\ e \end{bmatrix} = \begin{bmatrix} 0 & 0 & w_{ca+sc} \\ w_P G & -w_P & w_P G \\ G & -I & G \end{bmatrix} \begin{bmatrix} u_{ca+sc\Delta} \\ r \\ u \end{bmatrix} \tag{148}$$

where

$$P=\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \vdots & w_{ca+sc} \\ w_P G & -w_P & \vdots & w_P G \\ \cdots & \cdots & \cdots & \cdots \\ G & -I & \vdots & G \end{bmatrix} \tag{149}$$

and the description of the plant and controller is

$$N = P_{11} + P_{12}G_c(I - P_{22}G_c)^{-1}P_{21} \overset{\Delta}{=} F(P, G_c) \tag{150}$$

With the figure the generalized plant is illustrated in Figure 40.

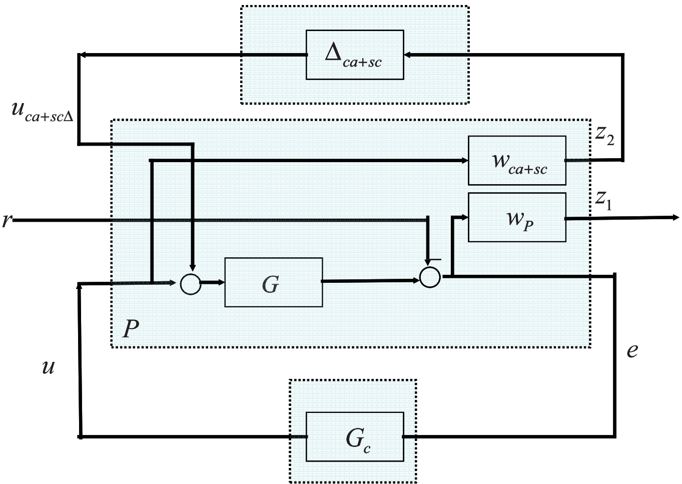

Figure 40. General control configuration for the case with uncertainty in transmission delay.

This general method for formulating control problems has been introduced by Doyle (1983).

7.2.2 Weight selection

In the model description developed in the previous section there are two design parameters: the performance weight w_P and the delay weight w_{ca+sc}. The weights are used to force the closed loop transfer functions, the sensitivity function $S = \dfrac{1}{1+L}$, the complementary sensitivity function $T = \dfrac{L}{1+L}$ and the loop transfer function

$L = G_c G$ to behave in a desired manner. These act as upper bounds for the desired performance; information about the upper bound delay will also be used in creating the weighting functions.

Delay weight w_{ca+sc}. The information on the upper bound network delay is utilized in creating the weighting functions w_{ca+sc}. The weight determination procedure consists of selecting the transfer function of a nominal plant $G(s)$ and determining at each frequency the smallest radius $l_{nw}(\omega)$ which includes all possible plant and network combinations Π:

$$l_{nw}(\omega) = \max_{G_{nw} \in \Pi} \left| \frac{G_{nw}(j\omega) - G(j\omega)}{G(j\omega)} \right| \tag{151}$$

and choosing the weight w_{ca+sc} so that:

$$\left| w_{ca+sc}(j\omega) \right| \geq l_{nw}(\omega), \forall \omega \tag{152}$$

For the SISO system, the smallest radius $l_{nw}(\omega)$ for all possible plant and network combinations where the actuator path delay and the sensor path delay are combined into a single multiplicative uncertainty weight can be presented as:

$$l_{nw}(\omega) = \max_{G_{nw} \in \Pi} \left| \frac{e^{-j\omega(\overline{\tau}_{ca} + \overline{\tau}_{sc})} G(j\omega) - G(j\omega)}{G(j\omega)} \right| = \max_{G_{nw} \in \Pi} \left| e^{-j\omega(\overline{\tau}_{ca} + \overline{\tau}_{sc})} - 1 \right| \tag{153}$$

where $\overline{\tau}_{ca}$ is the upper bound delay for the packet transmission in the actuator path, and $\overline{\tau}_{sc}$ the upper bound delay in the sensor path. Thus, for the frequencies below $\pi / (\overline{\tau}_{ca} + \overline{\tau}_{sc})$, this relative error is upper bounded by $\left| e^{-j\omega(\overline{\tau}_{ca} + \overline{\tau}_{sc})} - 1 \right|$ and, at the frequency $\pi / (\overline{\tau}_{ca} + \overline{\tau}_{sc})$ using the Euler identity ($e^{-j\pi} = -1$), we obtain 2 as the maximum of $l_{nw}(\omega)$:

$$l_{nw}(\omega) = \begin{cases} \left| e^{-j\omega(\overline{\tau}_{ca} + \overline{\tau}_{sc})} - 1 \right| & \omega < \pi / (\overline{\tau}_{ca} + \overline{\tau}_{sc}) \\ 2 & \omega > \pi / (\overline{\tau}_{ca} + \overline{\tau}_{sc}) \end{cases} \tag{154}$$

Rational transfer functions are usually preferred for $w_{ca+sc}(j\omega)$. The following weight is considered as a good approximation of the delay for the H_∞ synthesis (Wang *et al.* 1994):

$$w_{ca+sc}(s) = \frac{(\overline{\tau}_{ca} + \overline{\tau}_{sc}) \cdot s}{1 + (\overline{\tau}_{ca} + \overline{\tau}_{sc}) \cdot s / 3.465} \tag{155}$$

This weight was obtained experimentally from the complex Pade approximant by increasing its values at a higher frequency without changing the value at low frequencies.

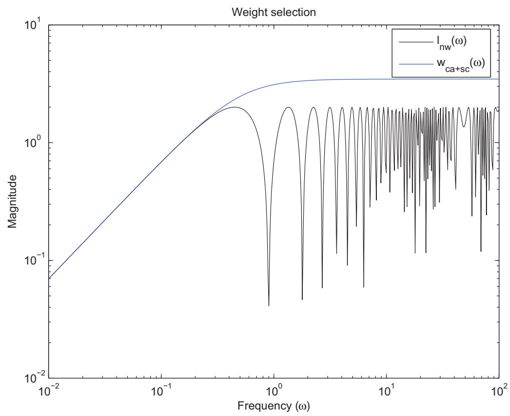

Figure 41. Delay weight $w_{ca+sc}(\omega)$ and $l_{nw}(\omega)$ for $(\overline{\tau}_{ca} + \overline{\tau}_{sc}) = 7$.

Integrator weight w_P. Weight w_P is related to the sensitivity function S. For good tracking accuracy in controlled outputs, the sensitivity function is required to be small. It suggests forcing integral action into the controller by selecting $\frac{1}{s}$ shape in the weights associated with the controlled outputs. However, a pure integrator cannot be included in w_P, since the standard H_∞ optimal control problem would not be well posed in the sense that the corresponding generalized plant could not be stabilized by the feedback controller. (Plant P becomes not invertible.) Therefore, the following approximation of an integrator was implemented as a weighting function w_P for the sensitivity function S:

$$w_P(s) = \frac{s/M + \omega_B}{s + \omega_B A} \tag{156}$$

where ω_B is the bandwidth for which the control is wanted to be effective, M is the desired maximum peak of S for desired robustness, and A is an arbitrary small number used to avoid numerical problems describing maximum steady state error. The weight is illustrated in Figure 42.

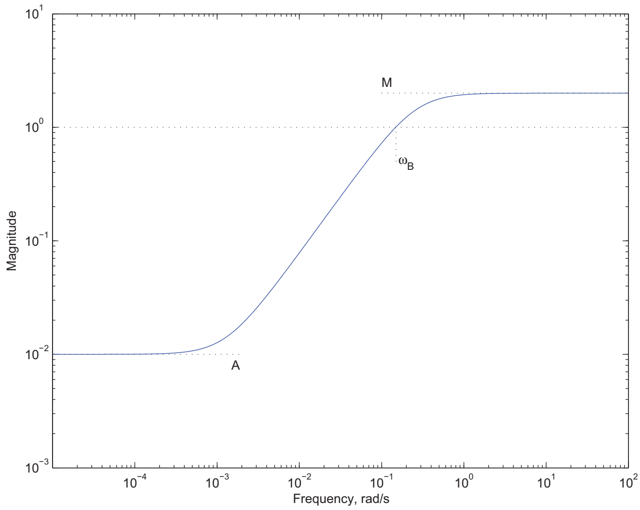

Figure 42. Weight for integrator w_P.

7.2.3 Controller synthesis

Nominal performance in terms of weighted sensitivity function can be given as follows (Skogestad and Postlethwaite, 2005):

$$\left|w_P S\right| < 1 \qquad \forall \omega \text{ or}$$
$$\left|w_P\right| < \left|1 + L\right| \qquad \forall \omega \tag{157}$$

For robust performance (RP), it is required that the above performance criteria are satisfied for all possible plants, including the worst case uncertainty:

$$\left|w_P S_{nw}\right| = \left|\frac{w_P}{1 + L_{nw}}\right| < 1 \qquad \forall \omega \tag{158}$$

In case of one multiplicative uncertainty $L_{nw} = L(1 + w_{ca+sc}\Delta_{ca+sc})$, considering that $|\Delta_{ca+sc}|$ is 1 under worst case conditions and pointing to the opposite direction of L, the RP condition can be simplified as follows:

$$\left|\frac{w_P}{1+L_{nw}}\right| = \left|\frac{w_P}{1+L(1+w_{ca+sc}\Delta_{ca+sc})}\right| \leq \left|\frac{w_P}{1+L-Lw_{ca+sc}}\right| < 1 \qquad \forall\omega \tag{159}$$

or

$$\begin{aligned} &|w_P| < |1+L-w_{ca+sc}L| \\ &\left|\frac{w_P}{1+L}\right| + \left|\frac{w_{ca+sc}L}{1+L}\right| = |w_P S| + |w_{ca+sc}T| < 1 \qquad \forall\omega \end{aligned} \tag{160}$$

Furthermore, noting that $|w_P S| + |w_{ca+sc}T|$ is a vector 1-norm and is therefore related to the vector 2-norm as follows:

$$|w_P S| + |w_{ca+sc}T| \leq \sqrt{2}\sqrt{|w_P S|^2 + |w_{ca+sc}T|^2} \tag{161}$$

condition $|w_P S| + |w_{ca+sc}T| < 1$ can be approximated closely as

$$\max_{\omega}\left(\sqrt{|w_P S|^2 + |w_{ca+sc}T|^2}\right) < \frac{1}{\sqrt{2}} \tag{162}$$

This is a standard H_∞ mixed sensitivity optimization problem where the condition has been slightly strengthened. Therefore, the controller for RP can be obtained by minimizing with respect to G_c the following H_∞ norm:

$$\min_{G_c}\left\|\begin{matrix} w_P S \\ w_{ca+sc}T \end{matrix}\right\|_\infty = \min\left\|P_{11} + P_{12}G_c(I - P_{22}G_c)^{-1}P_{21}\right\|_\infty = \gamma_{\min} \leq \frac{1}{\sqrt{2}} \tag{163}$$

7.3 Simulations for enhanced control of a NCS using delay estimates

7.3.1 Upper bound delay computation

Consider a networked control system consisting of a real time process and controller that are connected over a full duplex Ethernet switched network. The controller sends periodically control actions to the process and obtains information from the process, respectively. In addition to this basic communication, overload traffic that illustrates other activities of the network is sent over the network. The structure of the system is illustrated in Figure 43.

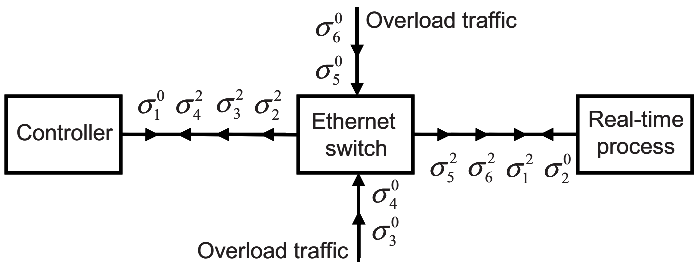

Figure 43. Structure of the network.

Six messages are sent periodically. The traffic sent from the process to the controller is given by $b_1^0(t)$, and the traffic from the controller to the process by $b_2^0(t)$, the background traffic from the other stations to the controller by $b_3^0(t), b_4^0(t)$, and the background traffic from the other stations to the process by $b_5^0(t), b_6^0(t)$. The arrival curves of the traffic are:

$$\begin{aligned} b_1^0(t) &= b_2^0(t) = \sigma_1^0 + \rho_1 t = 72 + 7200t \\ b_3^0 &= b_4^0(t) = b_5^0(t) = b_6^0(t) = \sigma_3^0 + \rho_3 t = 1526 + 305200t \end{aligned} \tag{164}$$

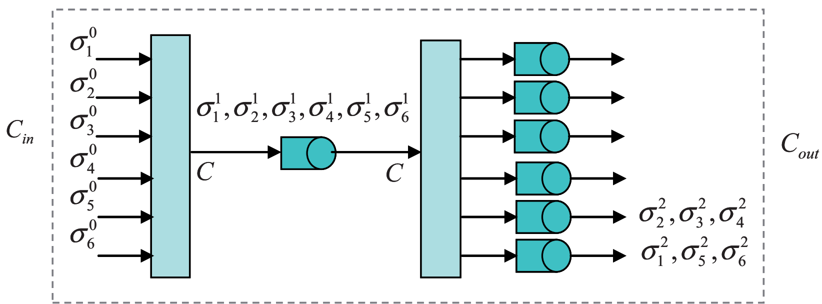

Figure 44. Model of a 6-port switch in a full duplex mode based on shared memory and a cut-through management.

The model of the Ethernet switch is illustrated in Figure 44. In order to compute end-to-end upper bounds for transmission delay for actuator and sensor paths (UBD_a and UBD_s), the route of each stream has to be first identified. Using Equations (136), (137) and (140) the following equations are obtained for burstiness values for crossing the multiplexer element, for streams $b_3^0, b_4^0(t), b_5^0(t), b_6^0(t)$ from perspective of flow source 1.

$$\begin{aligned}
\sigma_3^1 &= \sigma_3^0 + \rho_3 \overline{D}_{mux} = \sigma_1^1 + \rho_3 \frac{1}{C}\left(\sum_{z=1;z\neq 3}^{6} \left(\sigma_z^0 + \rho_z \left(\frac{\sigma_3^0}{(C_3-\rho_3)} + \frac{L_z}{C_z} \right) \right) + \frac{\sigma_3^0}{(C_3-\rho_3)}(C_3 - C) \right) \\
\sigma_4^1 &= \sigma_4^0 + \rho_4 \frac{1}{C}\left(\sum_{z=1;z\neq 4}^{6} \left(\sigma_z^0 + \rho_z \left(\frac{\sigma_4^0}{(C_4-\rho_4)} + \frac{L_z}{C_z} \right) \right) + \frac{\sigma_4^0}{(C_4-\rho_4)}(C_4 - C) \right) \\
\sigma_5^1 &= \sigma_5^0 + \rho_5 \frac{1}{C}\left(\sum_{z=1;z\neq 5}^{6} \left(\sigma_z^0 + \rho_z \left(\frac{\sigma_5^0}{(C_5-\rho_5)} + \frac{L_z}{C_z} \right) \right) + \frac{\sigma_5^0}{(C_5-\rho_5)}(C_5 - C) \right) \\
\sigma_6^1 &= \sigma_6^0 + \rho_6 \frac{1}{C}\left(\sum_{z=1;z\neq 6}^{6} \left(\sigma_z^0 + \rho_z \left(\frac{\sigma_6^0}{(C_6-\rho_6)} + \frac{L_z}{C_z} \right) \right) + \frac{\sigma_6^0}{(C_6-\rho_6)}(C_6 - C) \right)
\end{aligned} \tag{165}$$

or

$$\begin{aligned}
\sigma_3^1 &= \sigma_3^0 + \rho_3 \overline{D}_{mux} = \sigma_3^0 + \frac{\rho_3}{C}(\sigma_1^0 + \rho_1\left(\frac{\sigma_3^0}{(C_3-\rho_3)} + \frac{L_1}{C_1} \right) + \sigma_2^0 + \rho_2\left(\frac{\sigma_3^0}{(C_3-\rho_3)} + \frac{L_2}{C_2} \right) + \sigma_4^0 \\
&+ \rho_4\left(\frac{\sigma_3^0}{(C_3-\rho_3)} + \frac{L_4}{C_4} \right) + \sigma_5^0 + \rho_5\left(\frac{\sigma_3^0}{(C_3-\rho_3)} + \frac{L_5}{C_5} \right) + \sigma_6^0 + \rho_6\left(\frac{\sigma_3^0}{(C_3-\rho_3)} + \frac{L_6}{C_6} \right) + \frac{\sigma_3^0}{(C_3-\rho_3)}(C_3 - C))
\end{aligned} \tag{166}$$

$$\begin{aligned}
\sigma_4^1 &= \sigma_4^0 + \frac{\rho_4}{C}(\sigma_1^0 + \rho_1\left(\frac{\sigma_4^0}{(C_4-\rho_4)} + \frac{L_1}{C_1} \right) + \sigma_2^0 + \rho_2\left(\frac{\sigma_4^0}{(C_4-\rho_4)} + \frac{L_2}{C_2} \right) + \sigma_3^0 \\
&+ \rho_3\left(\frac{\sigma_4^0}{(C_4-\rho_4)} + \frac{L_3}{C_3} \right) + \sigma_5^0 + \rho_5\left(\frac{\sigma_4^0}{(C_4-\rho_4)} + \frac{L_5}{C_5} \right) + \sigma_6^0 + \rho_6\left(\frac{\sigma_4^0}{(C_4-\rho_4)} + \frac{L_6}{C_6} \right) + \frac{\sigma_4^0}{(C_4-\rho_4)}(C_4 - C))
\end{aligned} \tag{167}$$

$$\begin{aligned}
\sigma_5^1 &= \sigma_5^0 + \frac{\rho_5}{C}(\sigma_1^0 + \rho_1\left(\frac{\sigma_5^0}{(C_5-\rho_5)} + \frac{L_1}{C_1} \right) + \sigma_2^0 + \rho_2\left(\frac{\sigma_5^0}{(C_5-\rho_5)} + \frac{L_2}{C_2} \right) + \sigma_3^0 \\
&+ \rho_3\left(\frac{\sigma_5^0}{(C_5-\rho_5)} + \frac{L_3}{C_3} \right) + \sigma_4^0 + \rho_4\left(\frac{\sigma_5^0}{(C_5-\rho_5)} + \frac{L_4}{C_4} \right) + \sigma_6^0 + \rho_6\left(\frac{\sigma_5^0}{(C_5-\rho_5)} + \frac{L_6}{C_6} \right) + \frac{\sigma_5^0}{(C_5-\rho_5)}(C_5 - C))
\end{aligned} \tag{168}$$

$$\begin{aligned}
\sigma_6^1 &= \sigma_6^0 + \frac{\rho_6}{C}(\sigma_1^0 + \rho_1\left(\frac{\sigma_6^0}{(C_6-\rho_6)} + \frac{L_1}{C_1} \right) + \sigma_2^0 + \rho_2\left(\frac{\sigma_6^0}{(C_6-\rho_6)} + \frac{L_2}{C_2} \right) + \sigma_3^0 \\
&+ \rho_3\left(\frac{\sigma_6^0}{(C_6-\rho_6)} + \frac{L_3}{C_3} \right) + \sigma_4^0 + \rho_4\left(\frac{\sigma_6^0}{(C_6-\rho_6)} + \frac{L_4}{C_4} \right) + \sigma_6^0 + \rho_5\left(\frac{\sigma_6^0}{(C_6-\rho_6)} + \frac{L_5}{C_5} \right) + \frac{\sigma_6^0}{(C_6-\rho_6)}(C_6 - C))
\end{aligned} \tag{169}$$

As the values of $b_1^0(t), b_2^0(t)$ are lower than the values of $b_3^0, b_4^0(t), b_5^0(t), b_6^0(t)$, Equations (136), (138) and (140) should be used calculating the burstiness values for crossing the multiplexer. As the arrival curves for flows 3..6 are the same, it does not matter with respect to which the formulas are generated, i.e. which of three is chosen as a reference flow. With flow 3 as a reference flow the burstiness for streams σ_1^1 and σ_2^1 are:

$$\sigma_1^1 = \sigma_1^0 + \rho_1 \overline{D}_{mux} =$$
$$\sigma_1^1 + \rho_1 \frac{1}{C}\left(\sum_{z=1;z\neq 3}^{6} \left(\sigma_z^0 + \rho_z \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_z}{C_z} \right)\right) + \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} \right)(C_3 - C) - \rho_1 \frac{L_1}{C_1} + L_3 \right) \tag{170}$$

$$\sigma_2^1 = \sigma_2^0 + \rho_2 \frac{1}{C}\left(\sum_{z=1;z\neq 3}^{6} \left(\sigma_z^0 + \rho_z \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_z}{C_z} \right)\right) + \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} \right)(C_3 - C) - \rho_2 \frac{L_2}{C_2} + L_3 \right) \tag{171}$$

$$\sigma_1^1 = \sigma_1^0 + \rho_1 \overline{D}_{mux} = \sigma_1^0 + \frac{\rho_1}{C}\Bigg(\left(\sigma_1^0 + \rho_1 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_1}{C_1} \right)\right) + \left(\sigma_2^0 + \rho_2 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_2}{C_2} \right)\right) +$$
$$\left(\sigma_4^0 + \rho_4 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_4}{C_4} \right)\right) + \left(\sigma_5^0 + \rho_5 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_5}{C_5} \right)\right) + \left(\sigma_6^0 + \rho_6 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_6}{C_6} \right)\right)$$
$$+ \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} \right)(C_3 - C) - \rho_1 \frac{L_1}{C_1} + L_3 \Bigg) \tag{172}$$

$$\sigma_2^1 = \sigma_2^0 + \frac{\rho_2}{C}\Bigg(\left(\sigma_1^0 + \rho_1 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_1}{C_1} \right)\right) + \left(\sigma_2^0 + \rho_2 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_2}{C_2} \right)\right) +$$
$$\left(\sigma_4^0 + \rho_4 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_4}{C_4} \right)\right) + \left(\sigma_5^0 + \rho_5 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_5}{C_5} \right)\right) + \left(\sigma_6^0 + \rho_6 \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} + \frac{L_6}{C_6} \right)\right)$$
$$+ \left(\frac{\sigma_3^0}{(C_3-\rho_3)} - \frac{L_3}{C_3} \right)(C_3 - C) - \rho_2 \frac{L_2}{C_2} + L_3 \Bigg) \tag{173}$$

Using (136), (140) and (141) the burstiness value when crossing a queue element can be calculated. Note from the Figure 44 that there are two rows of queue elements, however as in the first row, both, input and output have the same capacity, C, the burstiness value remains unchanged. The following equations are obtained for all flows for crossing two FIFO queue element in the second row:

$$\sigma_1^2 = \sigma_1^1 + \rho_1 \overline{D}_{queue} =$$
$$\sigma_1^1 + \frac{\rho_1}{C_{out}} \frac{C - C_{out}}{C - (\rho_1 + \rho_5 + \rho_6)} (\sigma_1^1 + \sigma_5^1 + \sigma_6^1) \tag{174}$$

$$\sigma_2^2 = \sigma_2^1 + \frac{\rho_2}{C_{out}} \frac{C - C_{out}}{C - (\rho_2 + \rho_3 + \rho_4)} (\sigma_2^1 + \sigma_3^1 + \sigma_4^1) \tag{175}$$

$$\sigma_3^2 = \sigma_3^1 + \frac{\rho_3}{C_{out}} \frac{C - C_{out}}{C - (\rho_2 + \rho_3 + \rho_4)} (\sigma_2^1 + \sigma_3^1 + \sigma_4^1) \tag{176}$$

$$\sigma_4^2 = \sigma_4^1 + \frac{\rho_4}{C_{out}} \frac{C - C_{out}}{C - (\rho_2 + \rho_3 + \rho_4)} (\sigma_2^1 + \sigma_3^1 + \sigma_4^1) \tag{177}$$

$$\sigma_5^2 = \sigma_5^1 + \frac{\rho_5}{C_{out}} \frac{C - C_{out}}{C - (\rho_1 + \rho_5 + \rho_6)} (\sigma_1^1 + \sigma_5^1 + \sigma_6^1) \tag{178}$$

$$\sigma_6^2 = \sigma_6^1 + \frac{\rho_6}{C_{out}} \frac{C - C_{out}}{C - (\rho_1 + \rho_5 + \rho_6)} (\sigma_1^1 + \sigma_5^1 + \sigma_6^1) \tag{179}$$

Now, 12 equations have been obtained and there are 12 unknowns ($\sigma_1^1, \sigma_2^1, \sigma_3^1, \sigma_4^1, \sigma_5^1, \sigma_6^1, \sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2, \sigma_6^2$). The problem is linear and all unknowns can be solved.

Assume that the network links in the full-duplex mode are configured at $C_1 = C_2 = C_3 = C_4 = C_5 = C_6 = C_{out} = 10\text{Mb/s}$, maximum length of frames in flow 1 and 2 is $L_1 = L_2 = 72 \cdot 8 = 576$ bits, and maximum length of frames in flows 3..6 is $L_3 = L_4 = L_5 = L_6 = 1526 \cdot 8 = 12208$ bits, the capacity of the Ethernet switch according to specifications is $C = 40000000$ bits/s, burstiness values in traffic generators are $\sigma_1^0 = \sigma_2^0 = 72 \cdot 8 = 576$ bits, $\sigma_3^0 = \sigma_4^0 = \sigma_5^0 = \sigma_6^0 = 1526 \cdot 8 = 12208$ bits, average arrival rates are $\rho_1 = \rho_2 = 7200 \cdot 8 = 57600$ bits/s, $\rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5 = \rho_6 = 305200 \cdot 8 = 2441600$ bits/s. Inserting these to Equations (170-173), evaluating, and inserting the results to (174-179) all burstiness values can be determined. The following numerical values for unknown burstiness values were obtained: $\sigma_1^1 = \sigma_2^1 = 648.0$, $\sigma_3^1 = \sigma_4^1 = \sigma_5^1 = \sigma_6^1 = 12835.9$, $\sigma_1^2 = \sigma_2^2 = 777.8$, $\sigma_3^2 = \sigma_4^2 = \sigma_5^2 = \sigma_6^2 = 18334.8$

When the burstiness values are known, the end-to-end upper bound delay for stream 1 (controller to process path) and stream 2 (process to controller path) are calculated from Equation (145):

$$\bar{\tau}_{ca} = \frac{\sigma_1^2 - \sigma_1^0}{\rho_1} \approx \frac{777.8\text{bits} - 576\text{bits}}{57600\text{bits/s}} \approx 3.5\,\text{ms} \tag{180}$$

$$\bar{\tau}_{sc} = \frac{\sigma_2^2 - \sigma_2^0}{\rho_2} \approx \frac{777.8\text{bits} - 576\text{bits}}{57600\text{bits/s}} \approx 3.5\,\text{ms} \tag{181}$$

7.3.2 Simulations of the robust control design in the Matlab environment

Next, the upper bound delay was used in checking whether the system can become unstable under such delay. Consider again the system given by Equation (34). With the upper bound delay values calculated in the section above, the stability criteria given by Equation (14) can be calculated to check whether the system can become unstable under such delay. Resulting Bode diagram of the closed loop transfer function and the stability limit are shown in Figure 45. From the figure it can be seen that the frequency response curve of the closed loop transfer function and the stability limit curve intercept. That is, at frequencies between 0.1-9 rad/ms the stability criteria doesn't hold, and the feedback control loop can become unstable at this frequency range. Thus, a delay compensation strategy is needed.

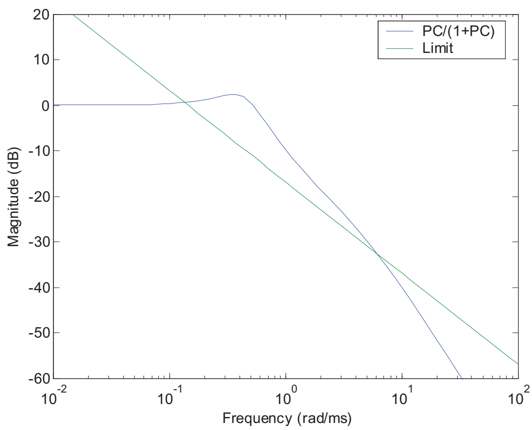

Figure 45. Stability checking for the control system and upper bound delay estimate using the criteria given.

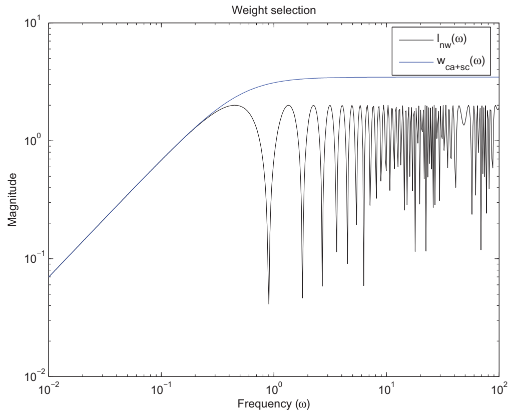

Figure 46. Weight selection for the network-induced delay ($UBD_1 = 3.5 \;\; UBD_2 = 3.5$).

The robust controller to compensate for the network transmission delay was obtained by solving the mixed sensitivity problem as presented in Equation (162). The weight presented in Equation (155) was chosen as a weighting function, w_{ca+sc}, for the complementary sensitivity function T. Figure 46 presents the frequency response of the obtained weight compared to the ideal case illustrated by Equation (154) when the upper bound delay for the actuator and sensor paths is 3.5 ms.

The following approximation of an integrator was implemented as a weighting function w_P for the sensitivity function S:

$$w_P(s) = \frac{s/M + \omega_B}{s + \omega_B A} = \frac{s/2 + 0.129}{s + 0.129 \cdot 0.001} \tag{182}$$

where ω_B is the bandwidth where control is effective, M is the desired maximum peak of ω_B, and A is an arbitrary small number used to avoid numerical problems. The H_∞ optimal controller for this mixed sensitivity problem was found using the Matlab™ Robust control toolbox. The following controller was obtained:

$$G_c(s) = \frac{17.01 \cdot s^3 + 96.88 \cdot s^2 + 60.8 \cdot s^2 + 8.421}{s^4 + 24.53 \cdot s^3 + 125.3 \cdot s^2 + 113.6 \cdot s^2 + 0.015} \tag{183}$$

The performance of the system under the robust controller is illustrated in Figure 47. The first column of the figure illustrates the simulated delay variation in the sensor part, the second column the delay variation in the actuator path, and the third column the response of the closed loop system. Rows 1-4 illustrate the performance of the robust controller under increased delay conditions. The result for the discrete system is presented by (34) where 0.02 ms was used as the sampling period. In the simulation of the network, the transmission delays over the network in the sensor and on the actuator side were assumed to vary randomly between zero and the upper delay value estimate. The same network delay variation used for evaluating the Smith predictor based compensation approach was implemented.

It can be concluded from the figure that the performance criteria are achieved when the delay is increased; in fact, the response of the process under the robust controller is almost completely insensitive to the increase in the delay. However, this approach provides relatively conservative results in terms of performance compared to the Smith predictor-based approach, especially when the overall network delay is short.

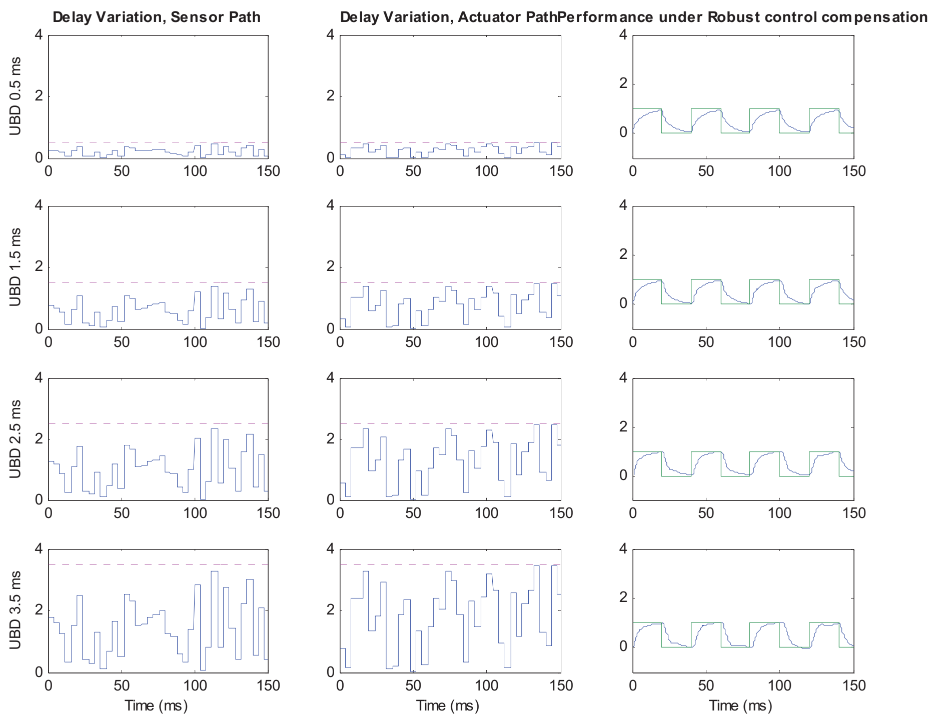

Figure 47. Performance of the robust controller during the increased delay.

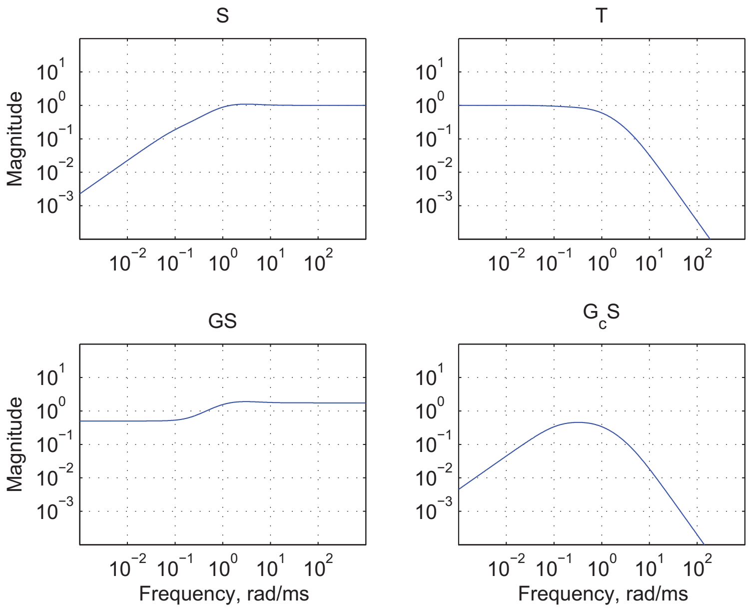

Figure 48. KS, S, GS and G_cS graphs for the obtained controller structure.

The achievements of performance specifications can also be seen from the frequency domain graphs S, T, GS, and G_cS in Figure 48.

7.3.3 Testing the robust control design in the experimental platform

The simulation result that reflects the second case obtained with the Matlab simulation presented in Figure 16 and Figure 47 is illustrated in Figure 49. The round trip time that occurred in the network is shown on the left side of Figure 49 and the corresponding process response on the right. According to Figure 49, the round trip time experienced by the controller is around 30 ms, which corresponds to the delay of 15 ms for the sensor and actuator paths or the 1.5 ms delay used in the Matlab simulations. Scaling was performed to better study the effect of the network and minimize other timing issues such as operation system performance. It can be concluded from the response curve of the process on the right side that the controller based on the robust control design is able to maintain the performance level of the system. The response time of the process is equal to 200 ms, and corresponds closely to the result obtained with the Matlab simulations (20 ms).

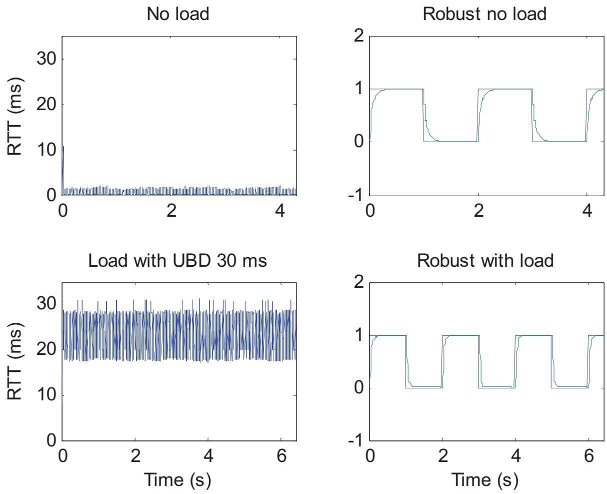

Figure 49. Performance of the robust controller under a varying network delay.

8 Enhanced control of a NCS using delay estimations and information of the measurement quality

The aim of this chapter is to propose a control approach for a NCS that considers not only time-varying effects induced by the network, but also uncertainty coming from inaccurate process measurements and modeling errors. The assumptions on the network are the same as in the previous chapter and the same method to obtain the upper bound end-to-end delay is employed. Another robust control approach is proposed that makes the controller tolerant to both the delay coming from the network as well as for uncertainties coming from the process and process measurement. First, a generalized plant with uncertainties in the model and measurements is constructed; second, a procedure to select weights for controller design is presented; third, the procedure for the controller synthesis is illustrated; and finally, the performance of the control scheme is illustrated with Matlab simulations.

8.1 Robust control design for delay and measurement error tolerance

In this section, a more general robust control-based design method for the network control system will be presented. In the previous section, the main assumptions can be often too optimistic. Also, here the end-to-end upper bound transmission delay will be utilized in the control design to obtain an estimate of the network performance.

8.1.1 Generalized plant with uncertainties in the model and measurements

The structure of the NCS system for which the controller synthesis will be proposed is presented in Figure 50. Compared to Figure 39, there is also now a measurement error around the plant G; further, the measurement and actuator path delays are not lumped. Lumping is not always possible, as in the case of a time variance, a lumped system is not the same.

First, a generalized plant is obtained by opening all loops in the system and writing down the signals. The input consists of control signals, reference input r, and perturbed signals $u_{ca\Delta}$, $y_{sc\Delta}$, and $y_{G\Delta}$. The output set consists of error signal e, and weighted exogenous outputs z_1, z_2, z_3, z_4. The following expression for the generalized plant *P* is obtained:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ e \end{bmatrix} = \begin{bmatrix} w_P & -w_P G & -w_P & -w_P & -w_P G \\ 0 & 0 & 0 & 0 & w_{ca} \\ 0 & w_{sc} G & 0 & w_{sc} & w_{sc} G \\ 0 & w_G G & 0 & 0 & w_G G \\ I & -G & -I & -I & -G \end{bmatrix} \begin{bmatrix} r \\ u_{ca\Delta} \\ y_{sc\Delta} \\ y_{G\Delta} \\ u \end{bmatrix} \tag{184}$$

which can be partitioned as

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} w_P & -w_P G & -w_P & -w_P & \vdots & -w_P G \\ 0 & 0 & 0 & 0 & \vdots & w_{ca} \\ 0 & w_{sc} G & 0 & w_{sc} & \vdots & w_{sc} G \\ 0 & w_G G & 0 & 0 & \vdots & w_G G \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I & -G & -I & -I & \vdots & -G \end{bmatrix} \tag{185}$$

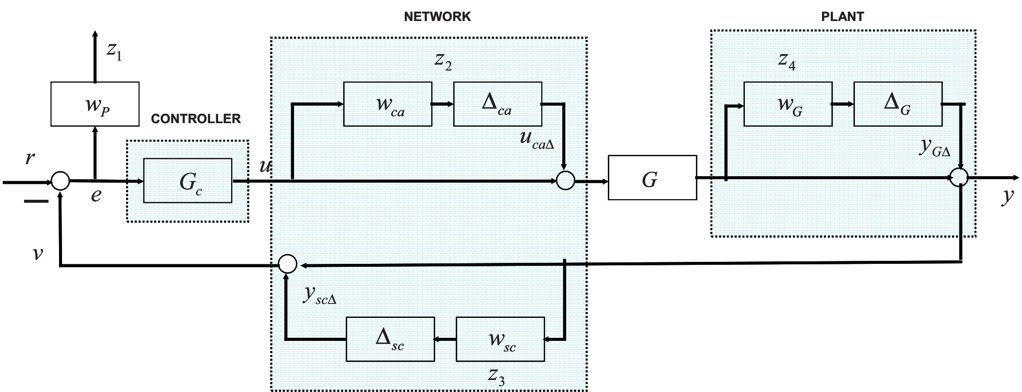

Figure 50. An approximated NCS for a robust controller synthesis with a measurement error.

Furthermore, a lower linear fractional transformation (LFT) is

$$N = P_{11} + P_{12}G_c(I - P_{22}G_c)^{-1}P_{21} \tag{186}$$

and an upper LFT is

$$F = F_u(N,\Delta) = N_{22} + N_{21}\Delta(I - N_{11}\Delta)^{-1}N_{12} \tag{187}$$

For the stability analysis, $M\Delta$ is needed where $M = N_{11}$. The general control configuration where the blocks related to uncertainties have been drawn out is represented in Figure 51.

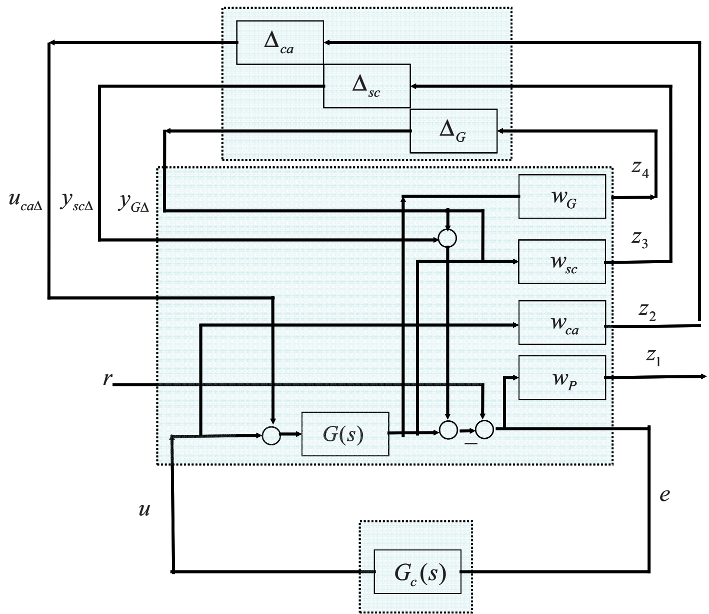

Figure 51. General control configuration for the NCS with uncertainties in transmission delay.

8.1.2 Weight selection

For the weights w_{ca}, w_{sc}, w_P, the same guidelines can be used as presented in Section 7.2.2. For the measurement uncertainty related weight w_G, a different dynamic uncertainty representation can be used. The goal is to first determine all sets of perturbed plants, and then to choose the weight so that it covers all perturbations. Consider that the measurement uncertainty comes from the uncertain values of gain k, time constant τ and plant delay θ. The set of possible plants can be represented as

$$G_P(s) = e^{-\theta_P} \frac{k_P}{1+\tau_P} G_0(s) \qquad k_{\min} \le k_P \le k_{\max}, \tau_{\min} \le \tau_P \le \tau_{\max}, \theta_{\min} \le \theta_P \le \theta_{\max} \tag{188}$$

where k_P is an uncertain gain, τ_P is uncertain time constant, θ_P is uncertain plant dead time and $G_0(s)$ is a transfer function with no uncertainties.

The weight w_G can be chosen by determining the frequency response of all perturbed plants and by choosing the weight so that it covers all variations. The choosing procedure can be started by trying the simple first order weight, increasing the order and then modifying the weight so that all perturbed plants are included.

8.1.3 Controller synthesis

After the weights have been defined, the controller synthesis problem can be solved with the μ-synthesis via DK-iteration procedure. Starting from the upper bound on μ in terms of the scaled singular value:

$$\mu(K) \le \min_{D \in \bar{D}} \bar{\sigma}(DND^{-1}) \tag{189}$$

the idea is to find the controller that minimizes the peak value over the frequency of this upper bound, or

$$\min_K (\min_{D \in \bar{D}} \left\| DN(K)D^{-1} \right\|_\infty) \tag{190}$$

by alternating $\left\| DN(K)D^{-1} \right\|_\infty$ with respect to either K (controller) or D (while holding the other fixed). To start, iteration $D(s)$ is chosen with an appropriate structure. The iteration continues until a satisfactory performance is achieved. For complex perturbations, the scaled singular value $\bar{\sigma}(DND^{-1})$ is a tight upper bound on $\mu(N)$ in most cases, and minimizing the upper bound $\left\| DN(K)D^{-1} \right\|_\infty$ forms the basis for the *DK*-iteration. The use of μ assumes the uncertain perturbations to be time invariant. However, it can be shown that when all uncertainty blocks in $\left\| DN(K)D^{-1} \right\|_{\infty}$ are full and complex, the upper bound provides a necessary and sufficient condition for robustness to arbitrary slow time-varying linear uncertainty (Poolla and Tikka, 1995). Furthermore, the use of constant D-scales (D is not allowed to vary with frequency) provides sufficient and necessary conditions for robustness to arbitrary fast time varying linear uncertainty (Shamma, 1994).

8.1.4 Simulations of the robust control design in the Matlab environment

Consider an example similar to that presented in Section 5.3.1. The nominal plant is

$$G = \frac{2}{(s+5)(s+0.2)} \tag{191}$$

and nominal PI controller is:

$$G_c = \frac{0.5508s + 0.4529}{s} \tag{192}$$

It is assumed that due to measurement and model uncertainties, in the real plant there are small dead time, gain and pole uncertainties:

$$G_d = e^{-\theta \cdot s} \frac{K}{(s+\tau)(s+0.2)} \tag{193}$$

where θ, K, and τ are set to vary in the following intervals:

$$\begin{aligned} \theta &= [\theta_{\min} \quad \theta_{nom} \quad \theta_{\max}] = [0.1 \quad 0.2 \quad 0.3] \\ K &= [K_{\min} \quad K_{nom} \quad K_{\max}] = [1.9 \quad 2 \quad 2.1] \\ \tau &= [\tau_{\min} \quad \tau_{nom} \quad \tau_{\max}] = [4 \quad 5 \quad 6] \end{aligned} \tag{194}$$

The phases in the controller synthesis include constructing the weights, formulating the problem, determining the bandwidth for control and solving the numerical problem.

Let us first determine the approximate bandwidth in the case of good control performance. In case the network delay is 3.5 ms in both directions, and a PI controller is used:

$$G_d = e^{-7s} \frac{2}{(s+5)(s+0.2)} \tag{195}$$

$$G_c = \frac{0.5508s + 0.4529}{s} \tag{196}$$

the step response of this system is illustrated in Figure 52 together with the sensitivity function $S = (1 + G_c G_d)^{-1}$ of this closed loop system. It can be seen from the figure that the frequency at which the sensitivity function crosses -3dB from below, the bandwidth w_b is around 0.02.

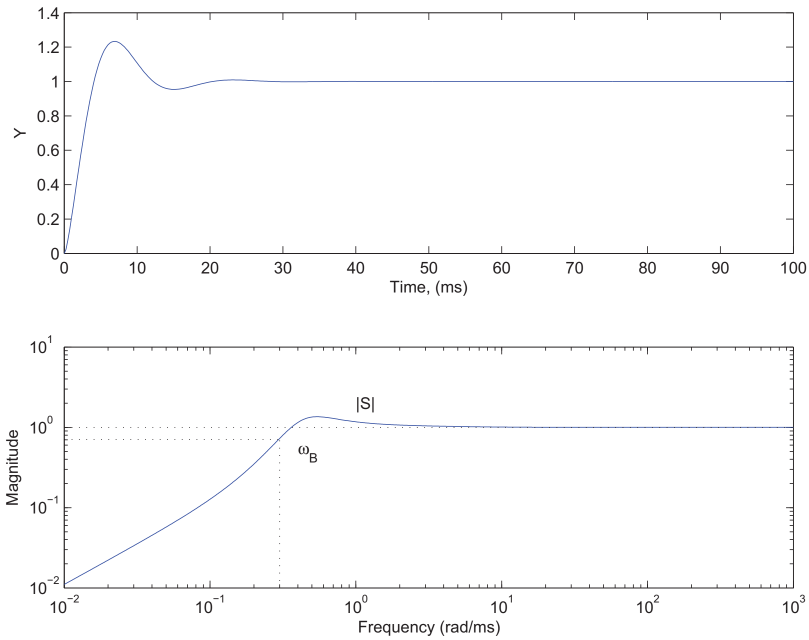

Figure 52. The step response of a nominal plant with a constant network delay and PI controller. The sensitivity function of the closed loop and desirable bandwidth.

In determining the weight w_I the frequency of perturbed plants G_p is plotted

$$G_d = e^{-\theta \cdot s} \frac{K}{(s+\tau)(s+0.2)} \tag{197}$$

where θ, K, and τ are set to vary in the following intervals:

$$\begin{aligned}
&\theta = [\theta_{\min} \quad \theta_{nom} \quad \theta_{\max}] = [0.1 \quad 0.2 \quad 0.3] \\
&K = [K_{\min} \quad K_{nom} \quad K_{\max}] = [1.9 \quad 2 \quad 2.1] \\
&\tau = [\tau_{\min} \quad \tau_{nom} \quad \tau_{\max}] = [4 \quad 5 \quad 6]
\end{aligned} \tag{198}$$

Choosing the performance weight can be started from the following second order weight:

$$w_G = \frac{(1+\frac{r_k}{2})\theta_{\max}s + r_k}{\frac{\theta_{\max}}{2}s+1} \cdot \frac{(\frac{\theta_{\max}}{2.363})^2 s^2 + 2\cdot 0.838 \cdot \frac{\theta_{\max}}{2.363}s+1}{(\frac{\theta_{\max}}{2.363})^2 s^2 + 2\cdot 0.685 \cdot \frac{\theta_{\max}}{2.363}s+1} \tag{199}$$

where r_k is relative magnitude of the gain uncertainty or

$$r_k = \frac{(k_{\max} - k_{\min})/2}{\bar{k}} \tag{200}$$

and $\bar{k}$ is average gain:

$$\bar{k} = \frac{k_{\min} + k_{\max}}{2} \tag{201}$$

As can be seen from Figure 53, this weight covers all high frequencies well; however, for lower frequencies an additional factor is required. By adding the additional factor 0.3, the following weight is obtained, which covers all perturbed plants

$$w_G = \frac{\left(((1+\frac{r_k}{2})\theta_{\max}s + r_k\right)\left(\left(\frac{\theta_{\max}}{2.363}\right)^2 s^2 + 2\cdot 0.838\cdot \frac{\theta_{\max}}{2.363}s+1\right)+0.3}{\left(\frac{\theta_{\max}}{2}s+1\right)\left(\left(\frac{\theta_{\max}}{2.363}\right)^2 s^2 + 2\cdot 0.685\cdot \frac{\theta_{\max}}{2.363}s+1\right)} \tag{202}$$

With this weight and the weights by choosing the weight for the actuator and sensor delays as presented in Section 7.2.2, a controller synthesis problem can be formulated and solved with a DK-iteration algorithm using a Matlab's μ-synthesis toolbox. With the bandwidth of 0.3, the obtained order was a high order of 21, while the γ value with obtained controller was 0.4.

The order of the controller is relatively high, by reducing the third order Hankel singular value reduction, the following controller was obtained:

$$G_c = \frac{-0.127s^2 + 17.74s + 4.154}{s^3 + 9.108s^2 + 32s + 0.0001} \tag{203}$$

The performance when there is a measurement uncertainty is illustrated in Figure 54. It can be seen from the figure that the performance is relatively good, and the controller is able to handle the measurement and model uncertainties.

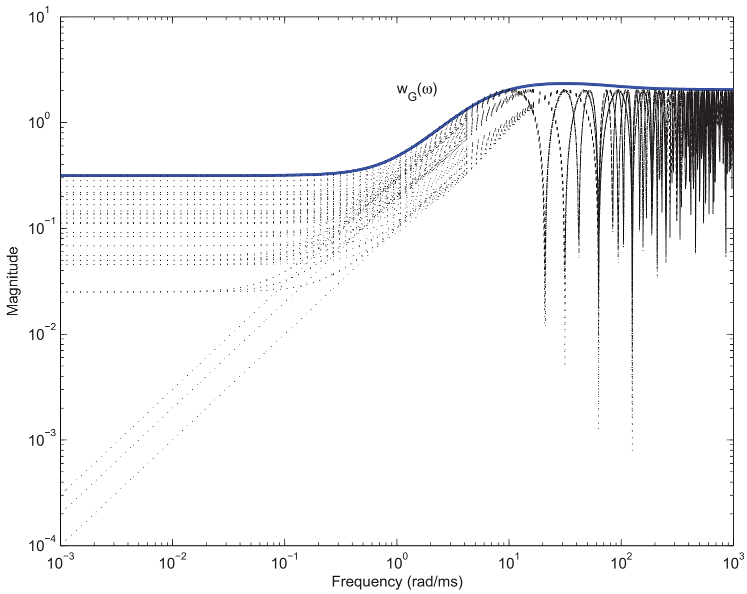

Figure 53. Weight selection for the NCS with uncertainty in the delay and process model.

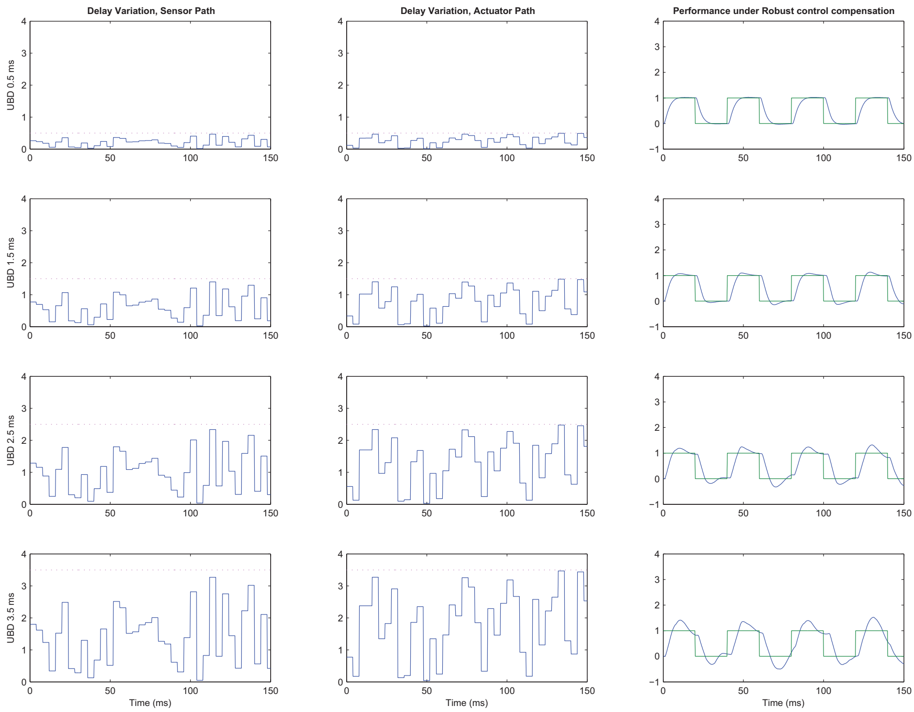

Figure 54. Performance for the robust controller under varying delay and measurement errors.

9 Summary of the results

This chapter contains a summary of the control strategies used in this thesis. The presented strategies are compared in terms of the required network information, accuracy of the network information, the applicability of control in a time varying environment, implementation and applicability.

In the simulations in the previous sections, the control strategies were compared to a traditional PI controller, the proportional gain and time constant of which were obtained by minimizing the integral of the squared error signals. From the comparisons the proposed control strategies seemed to outperform the PI controller under increased delay variation conditions. However, there are differences in the way the methods achieve this performance improvement and perform under abnormal conditions.

The time invariant Smith predictor presented in Section 5.2 can be used as a first step in performance improvement of a NCS when a traditional controller is inadequate. The benefits of the approach, in addition to its relative simplicity, is that the method is well suited to directly incorporate the delay measurement and can be considered as a well established technique to compensate for the dead time. The disadvantages of the method are its lack of robustness, its sensitivity to modeling errors, and its inapplicability to unstable systems. Because of the first and second disadvantages, when implementing the Smith predictor an adaptation procedure needs to be developed and should be applied when the measurement error or modeling error increases. As an example, the gain adaptation method presented for a nominal controller could be considered as first choice. Inapplicability to unstable stable system, even though it is worthwhile to consider in theory, is seldom an issue in practical applications since there are not many practical inherently unstable systems that are controlled over a network. There can obviously be many applications in which the set point is sent over the network to a local controller placed in the vicinity of inherently unstable process. In this case, the performance is not affected by the network if the set point changes are not frequent.

An issue that the time variant Smith predictor is not able to handle is the situation where the delay varies frequently. The transmission delay variation makes the overall system nonlinear and its subsystems become noncommutative. This can be considered as such a severe problem that it needs to be considered explicitly. The time variant Smith predictor developed in Section 6.2.5 using the polynomial systems theory as a tool is able to handle this. The disadvantages of the time variant Smith predictor related to the lack of robustness and modeling errors remain the same as in the time-invariant Smith predictor and those to be considered explicitly. Another disadvantage for practical applications comes from relative complexity of the calculations even though there are now more and more tools available.

A robust controller was developed in Section 7.2.3. The motivation for this is that it is not always possible to implement such instrumentation to the network so that the delay measurement is available. The availability of the delay measurement is a crucial requirement for Smith predictor type approaches. When the delay measurement is not obtainable, an estimate of the network performance should be obtained with some network model. The delay estimation in non-deterministic networks is known to be a challenging task, mainly due to the unpredictability of future network traffic. In this work, a network calculus-based network model was used, which gave the estimate of the upper bound transmission delay based on the investigation of the structure of the network and limits posed by the transmission capacity of the network devices. As an output, the model gave an upper bound network delay estimate that was subsequently used in a robust controller synthesis. The main advantages is that the method is well suited in incorporating the upper bound delay as the robust controller is formulated based on worst case performance. There is also no need to develop an adaptation procedure since all possible perturbations should be included in design phase. The disadvantage of the approach is that the controller is conservative when the level of traffic is low.

A controller was presented in Section 8.1.3 that handled all possible error sources in NCSs. The advantage of the method is in the stability of the overall system while the disadvantage is again the conservativeness at a low network traffic level. Also, as discussed in the section, another disadvantage is that the controller may not perform in case the transmission delay is varying rapidly, even though there are some exceptions when the uncertainty blocks in $\left\| DN(K)D^{-1} \right\|_{\infty}$ are full and complex or D-scales are frequency independent (constants), as discussed in the previous section.

The performance of the presented strategies under different conditions is summarized in Table 6 below. In the table, "μ-optimized" refers to a μ-optimized controller presented in Section 8.1.3; RobCon refers to the robust controller presented in Section 7.2.3; PolySysSP in Section 6.2.5; and SP and PI in Section 5.2.2, respectively. The order has been obtained based on the experiments carried out on the controllers and by examining the effect on the control performance in the development stage. The numbers from 1 to 5 indicate the suitability of the approach with respect to each other, 1 indicating the most suitable case.

Table 6. Summarized performance of different controllers in a comparative study.

	PI control	SP-control	Polynom. SP control	Robust control.	μ-optimized control
No delay information available	3	5	4	2	1
Delay measurement with error	5	2	1	3	4
Rough delay estimate of from a network model	3	5	4	1	2
Good measurement	5	2	1	3	4
Slowly varying delay	4	5	1	3	2
Rapidly varying delay	4	5	1	3	2
Applicability to applications	3	2	1	4	5
Computation synthesis complexity	5	4	1	3	2

10 Conclusions

In this work, delay compensation methods for Networked Control Systems were proposed while considering the protocol specifications of the switched Ethernet. The proposed approaches were classified into two categories depending on the information available from the network on its performance. In the first category, the measured information is used to enhance the overall performance of the controlled application; in the second case, the estimated information obtained with a detailed network model is utilized for the performance improvement.

For the approach with the use of measured information, two control compensation strategies are presented that differ with respect to their applicability to the control of time-varying systems. The first takes the time variant behavior into account by adjusting the controller parameters based on the uncertainty bounds introduced by the time variance. For this case, a Smith predictor-based approach was proposed. In the second case, time variant control methods are applied directly using the polynomial systems theory as a control design tool. In the estimated information approach, the models for the network are formed with the network calculus theory, the outputs of which are used in control design.

The proposed compensation strategies were evaluated with simulations made possible to further reduce the nondeterministic effect of the switched Ethernet, making the protocol more suitable for the control applications.

From the theoretical perspective, the tools that have been applied in the thesis to the control of networked systems are relatively well developed, and for the future, the development task includes reducing the effect of assumptions. One clear assumption that has been made is related to no data loss in communication. Even though the data loss can be modeled relatively well, as an increased time variant delay, this representation is not exact. In the case where the estimated information has been used, the main restriction is related to the estimate, which is relatively conservative, in the case where there is only little time-critical traffic. Also, the robust control method that has been applied often gives conservative results. In the polynomial system control approach, only simple problems can be solved due to a lack of computation tools for polynomial systems, especially in the MIMO case.

Future research related to the control of networked control systems should be more application driven or oriented. Until recently, there has been much theoretical research on NCSs with several approaches being developed with different assumptions concerning the communication networks and application requirements. From the theoretical perspective, many NCS challenges are understood and can be overcome. However, one of the main drawbacks is the lack of application on a NCS, even though the research area was originated as a possible need in the application area. While several small, prototype-scale applications have been made, only few small-scale industrial applications exist.

References

ANSI/ISA-95.00.01-2000. 2000. Enterprise-Control System Integration. American National Standard Institute/International Society of Automation.

AFNOR NF C46-603. 1990. FIP Bus for exchange of information between transmitters, actuators and programmable controllers, French Association for Standardizations.

Bauer, P. H., Sichitiu, M. L. & Premaratne, K. (2001). On the nature of the time-variant communication delays. In IASTED Conference on Modeling, Identification and Control, Innsbruck, Austria, 792–797.

Blomberg H. & Ylinen R. (1983). Algebraic theory for multivariable linear systems, Academic Press, London, 360 p.

Bosch CAN specification version 2.0. 1992. Robert Bosch GmbH, Germany.

Boudec Le, J.-Y., & Thiran P. (2001). Network calculus, a theory of deterministic queueing systems for the Internet, In: Lecture Notes in Computer Science, Springer Verlag, New Jersey.

Cao J., Zhong, S. & Hu, Y. (2008). Novel delay-dependent stability conditions for a class of MIMO networked control systems with nonlinear perturbation, Applied Mathematics and Computation, 197(2), 797–809.

CENELEC EN 50170. 1996. General purpose field communication system. European Commitee for Electrotechnical Standardization.

CENELEC EN 50170. 1996. General purpose field communication system Fieldbus. Volume 2: PROFIBUS. European Commitee for Electrotechnical Standardization.

CENELEC EN 50170. 1996. General purpose field communication system Fieldbus. Volume 2: PROFINET. European Commitee for Electrotechnical Standardization.

CENELEC EN 50254. 1996. High efficiency communications subsystems for small data packages. European Commitee for Electrotechnical Standardization.

CENELEC EN 50295. 1999. Low-voltage switchgear and control gear. Controller and device interface systems. Actuator Sensor interface (AS-i). European Commitee for Electrotechnical Standardization.

CENELEC EN 50325-4. 2002. Industrial communication subsystem based on ISO 11898 (CAN) for controller-device interfaces. European Commitee for Electrotechnical Standardization.

Chow, M.Y. & Tipsuwan, Y. (2003). Gain adaptation of networked DC motor controllers based on QoS variations, IEEE Transactions on Industrial Electronics, 50(5), 936–943.

Chen, W.H., Guan, Z.H. & Lu, X.M. (2004). Delay-dependent output feedback guaranteed cost control for uncertain time-delay systems, Automatica, 40(7), 1263–1268.

Chen, C.H., Lin, C.L. & Hwang, T.S. (2007). Stability of networked control systems with time-varying delays, IEEE Communications Letters, 11(3), 270-272.

Cheng, T.M. & Savkin, A.V. (2007). Output feedback stabilization of nonlinear networked control systems with non-decreasing nonlinearities: A matrix inequalities approach, International Journal of Robust and Nonlinear Control, 17(5), 387-404.

Cohn, P.M (1971). Free rings and their relations, Academic Press, London and New York, 346p.

ControlNet specification, release 2.0. 1997. ControlNet International.

Cruz, R. (1991). A calculus for network delay, part I: Network elements in isolation. IEEE Transactions on Information Theory, 37, 114–131.

DeviceNet specification - release 2.0, Vol. I and II. 1997. Open DeviceNet Vendor Association.

DIN 19245-1 to 19245-3. 1990. Process Fieldbus, PROFIBUS. German Institute for Standardization.

[DIN 19258-1 (1996)] DIN 19258-1. 1996. Interbus-S, sensor and actuator network for industrial control systems. German Institute for Standardization.

Doyle, J.C. (1983). Synthesis of robust controllers and filters, In Proceedings of IEEE Conference on Decision and Control, San Antonio, Texas, 109–114.

DS 21906. 1990. P-Net, Multi-master, multi-net fieldbus for sensor, actuator and controller communication. Danish Standards.

Eriksson, L. (2008). PID controller design and tuning in networked control systems, Doctoral dissertation, Helsinki University of Technology, Department of Automation and Systems Technology, Yliopistopaino, Helsinki.

Enhanced media access control with LonTalk protocol. 1995. Echelon Corporation.

Felser, M. (2005). Real-time Ethernet - industry prospective, Proceedings of the IEEE, 93(6), 1118-1129.

Fridman, E., Seuret, A. & Richard, J.P. (2004). Robust sampled-data stabilization of linear systems: an input delay approach, Automatica, 40(8), 1441-1446.

Gaderer G., Loschmidt, P., & Sauter, T. (2005). IEEE 1588 real-time networks with hybrid master group enhancements. In 4th International Workshop on Real.Time Networks (RTN'05), Palma de Mallorca, Spain.

Gao, H. & Chen, T. (2008). Network-based H_∞ output tracking control, IEEE Transactions on Automatic Control, 53(3), 655-667.

Gao, H., Chen, T. & Lam, J. (2008). A new delay system approach to network-based control, Automatica 44 (1), 39 – 52.

Georges, J.P., Divoux, T. & Rondeau, E. (2005). Confronting the performances of a switched Ethernet network with industrial constraints by using the network calculus, International Journal of Communication Systems 18, 877–903.

Georgiev, D. & Tilbury, D.M. (2006). Packet-based control: The H_2-optimal solution, Automatica, 42 (1), 137 – 144.

Gu, K., Kharitonov, V. L. & Chen, J. (2003). Stability of time-delay systems, Birkhhäuser, Boston, 354 p.

Hristu-Varsakelis, D. & Levine W. S. (2005). Handbook of networked and embedded control systems, Birkhäuser, Boston, 822 p.

Huang, D. & Nguang, S.K. (2008). State feedback control of uncertain networked control systems with random time delays, IEEE Transactions on Automatic Control, 53(3), 829-834.

Hristu-Varsakelis, D. & Zhang L. (2008). LQG control of networked control systems with access constraints and delays, International Journal of Control, 81(8), 1266-1280.

Hu, L.S., Bai, T., Shi, P. & Wu, Z. (2007). Sampled-data control of networked linear control systems, Automatica, 43(5), 903 – 911.

Huang, J., Wang, Y., Yang, S.H. & Xu, Q. (2009). Robust stability conditions for remote SISO DMC controller in networked control systems, Journal of Process Control, 19 (5) 743–750.

IEC 61784-1. 2007. Industrial communication networks - profiles - part 1: Fieldbus profiles. International Electrotechnical Commission.

IEC 61784-2. 2007a. Industrial communication networks - profiles - part 2: additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3. International Electrotechnical commission.

IEC 61784-2. 2007b. Industrial communication networks - profiles - part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3 IEC, CPF 6, Interbus. International Electrotechnical Commission.

IEC 61784-2. 2007c. Industrial communication networks - profiles - part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3 IEC, CPF 10, VNET/PI (Virtual Network protocol). International Electrotechnical Commission.

IEC 61784-2. 2007d. Industrial communication networks - profiles - part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3 IEC, CPF 11, TCnet. International Electrotechnical Commission.

IEC 61784-2. 2007e. Industrial communication networks - profiles - part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3 IEC, CPF 13, EPL (Ethernet PowerLink), Interbus. International Electrotechnical Commission.

IEC 65C/358/NP. 2004. Real-Time Ethernet: SERCOS III: Proposal for a Publicly Available Specification for Real-Time Ethernet, International Electrotechnical Commission.

IEEE 802.3-2000. 2000. IEEE Standard for information technology - specific requirement - part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. Institute of Electrical and Electronics Engineers.

IEEE 1588-2002. 2002. Standard for a precision clock synchronization protocol for networked measurement and control systems. Institute of Electrical and Electronics Engineers.

Imer, O.C., Yuksel, S., & Basar, T. (2006). Optimal control of LTI systems over unreliable communication links, Automatica, 42(9), 1420-1439.

Ishii, H. (2008). H_∞ control with limited communication and message losses, Systems and Control Letters, 57(4), 322-331.

ISO 11898. 1993. Road vehicle - interchange of digital information - Controller Area Network (CAN) for high-speed communication. International Organization for Standardization.

ISO/IEC 8802-3. 2000. Information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications. International Organization for Standardization/International Electrotechnical Commission.

Jasperneite, J., Shebab, K., & K. Weber (2004). Enhancement to the time synchronisation standard IEEE-1588 for a system of cascaded bridges. In Proceedings of the 5th IEEE International Workshop on Factory Communication Systems (WFCS'04), Vienne, 39-44.

Jiang, X. & Han, Q.A. (2008). New stability criteria for linear systems with interval time-varying delay, Automatica, 44(10), 2680–2685.

Jing, X.J., Tan, D.L. & Wang, Y.C. (2004). An LMI approach to stability of systems with severe time-delay, IEEE Transactions on Automatic Control, 49(7), 1192–1195.

Juanole G. &. Mouney, G. (2006). Real time distributed systems: QoS and impact on the performances of process control application, In Proceeding of the 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS2006), Kyoto, 1739-1746.

Kamen, E.W. (1988). The poles and zeros of a linear time-varying system, Linear Algebra and Its Applications, 98(1), 263-289.

Kao, C.Y. & Lincoln, B. (2004). Simple stability criteria for systems with time-varying delays, Automatica, 40(8), 1429-1434.

Kim, D.S., Lee, Y.S., Kwon, W.H. & Park, H.S. (2003). Maximum allowable delay bounds of networked control systems, Control Engineering Practice, 11(11), 1301–1313.

Kim, W.J., Ji, K. & Ambike, A. (2006). Networked real-time control strategy dealing with stochastic time delays and packet losses, Journal of Dynamic Systems, Measurement, and Control, 128(3), 681-685.

Lee, K.C., Lee, S. & Lee, M.H. (2006). Worst case communication delay of real-time industrial switched Ethernet with multiple levels, IEEE Transactions on Industrial Electronics, 53(5), 1669-1676.

Lee (2005), [Lee *et al.* (2005)] Lee, K.C., Lee, S. & Lee, M.H. (2005). QoS-based remote control of networked control systems via Profibus token passing protocol, IEEE Transactions on Industrial Informatics, 1(3), 183-191.

Li, J.G., Yuan, J.Q. & Lu, J.G. (2010). Observer-based H_∞ control for networked nonlinear systems with random packet losses, ISA Transactions, 49(1), 39-46.

Li, J.N., Zhang, Q.L., Wang, Y.K. & Cai, M. (2009). H_∞ control of networked control systems with packet disordering, IET Control, Theory and Applications, 3(11), 1463-1475.

Lian, F.L., Moyne, J.R. & Tilbury, D.M. (2001). Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet, IEEE Control Systems Magazine, 21(1), 66–83.

Lian, F.L., Moyne, J.R. & Tilbury, D. (2003). Modelling and optimal controller design of networked control systems with multiple delays, International Journal of Control, 76(6), 591–606.

Lin, C.L. & Chen, C.H. (2006). Robust control of interconnected network schemes subject to random timevarying communication delays, International Journal of Electronics and Communication, 60(9), 647 – 658.

Liou, L.W. & Ray, A. (1991). A stochastic regulator for integrated communication and control systems: Part I - Formulation of control law, Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control, 113(2), 604–611.

Liu, G.P., Mu, J.X., Rees, D. & Chai, S.C. (2006). Design and stability analysis of networked control systems with random communication time delay using the modified MPC, International Journal of Control, 79(4), 288–297.

Liu, G.P, Xia, Y., Rees, D. & Hu, W. (2007). Design and stability criteria of networked predictive control systems with random network delay in the feedback channel, IEEE

Transactions on Systems, Manufacturing, and Cybernetics—Part C: Applications and Reviews, 37(2), 173-184.

Ma, C., Chen, S. & Liu, W. (2007). Maximum allowable delay bound of networked control systems with multi-step delay, Simulation Modelling Practice and Theory 15(5), 513–520.

Matveev, A.S. & Savkin, A.V. (2004). Optimal design of networked control systems: computer control via asynchronous communication channels, International Journal of Control, 77(16), 1426–1437.

Michiels, W., Van Assche, V. & Niculescu, S.I. (2005). Stabilization of time-delay systems with a controlled time-varying delay and applications, IEEE Transactions on Automatic Control, 50(4), 493–504.

Morari, M. & Zafiriou, E. (1989). Robust process control, Prentice Hall, New Jersey, 512 p.

Mills, D.M. (1991). Internet time synchronisation: the network time protocol. IEEE Transactions on Automatic Control, 49(10), 1650-1667.

Montestruque, L.A., & Antsaklis, P. (2004). Stability of model-based networked control systems with time-varying transmission times, IEEE Transactions on Automatic Control, 49(9), 1562–1572.

Montestruque, L.A. & Antsaklis, P.J. (2007). Static and dynamic quantization in model-based networked control systems, International Journal of Control, 80(1), 87-101.

Niculescu, S.I. (2001). Delay effects on stability—a robust control approach, Springer, London, 383 p.

Nilsson, J. (1998). Real-time control systems with delays, Ph.D. dissertation, Department of Automatic Control, Lund Institute of Technology, Lund.

O'Brien, R.T.Jr. & Iglesias, P.A. (2001). On the poles and zeros of linear, time-varying systems, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 48(5), 565 - 577.

Open Modbus/TCP specification, Release 1.0. 1999. Modicon.

Otanez, P.G., Moyne, J.R. & Tilbury, D.M. (2002). Using deadbands to reduce communication in networked control systems, In Proceedings of American Control Conference (ACC02), Anchorage, 4, 3015–3020.

Peng, C. & Tan, Y.C. (2009). Delay-dependent robust H_∞ control for uncertain systems with time-varying delay, Information Sciences, 179(18), 3187-3197.

Park, H.S., Kim, Y.H., Kim, D.S. & Kwon, W.H. (2002). A scheduling method for network-based control systems, IEEE Transactions on control systems technology, 10(3), 319-330.

Poolla, K. & Tikku, A. (1995). Robust performance against time-varying structured perturbations, IEEE transactions on Automatic Control, 40(9), 382-387.

Richard, J.P. (2003). Time-delay systems: an overview of some recent advances and open problems, Automatica, 39(10), 1667–1694.

Sanfridson, M. (2000). Timing problems in distributed real-time computer control systems, Technical Report, Royal Institute of Technology, KTH, Stockholm.

Seifert, R. (2000). The Switch Book: The Complete Guide to LAN Switching Technology, Wiley; New York, 698 p.

Shamma, J. S., (1994). Robust stability with time-varying structured uncertainty, IEEE Transactions on Automatic Control, 39(4), 714-724.

Shousong, H. & Qixin, Z. (2003). Stochastic optimal control and analysis of stability of networked control systems with long delay, Automatica, 39(11), 1877-1884.

Skogestad, S. & Postlethwaite, I. (2005). Multivariable feedback control - analysis and design, 2nd Edition, Wiley, West Sussex, 608 p.

Specification of the TTP/C protocol. 1996. TTTech Computertechnik AG.

Srinivasagupta, D., Schättler, H. & Joseph, B. (2004). Time-stamped model predictive control: an algorithm for control of processes with random delays, Computers and Chemical Engineering, 28 (8), 1337–1346.

Tang, B., Liu, G.P. & Gui, W.H. (2008). Improvement of state feedback controller design for networked control systems, IEEE Transactions on Circuits and Systems - II: Express Briefs, 55(5), 464-468.

Tang, P.L. & de Silva, C.W. (2006). Compensation for transmission delays in an Ethernet-based control network using variable-horizon predictive control, IEEE Transactions on Control Systems Technology, 14(4), 707-718.

Tian, Y.C. & Levy, D. (2008). Compensation for control packet dropout in networked control systems, Information Sciences, 178(5), 1263–1278.

Tipsuwan, Y. & Chow, M.Y. (2003). Control methodologies in networked control systems, Control Engineering Practice, 11(10), pp. 1099–1111.

Vatanski, N. Ylinen, R. & Jämsä-Jounela, S.L. (2008). New time variant control system design for networked control systems, In Proceedings of the 18th International Symposium on Mathematical Theory of Networks and Systems (MTNS2008), Blackburg.

Vatanski, N., Georges, J.P., Aubrun, C., Rondeau, E. & Jämsä-Jounela, S.L. (2009). Networked control system with delay measurement and estimation. Control engineering practice, 16(11), 1321-1328.

Walsh, G.C., Beldiman, O., & Bushnell, L.G. (2001). Asymptotic behavior of nonlinear networked control systems, IEEE Transactions on Automatic Control, 46(7), 1093–1097.

Walsh, G.C., Beldiman, O., & Bushnell, L.G. (2002a). Error encoding algorithms for networked control systems, Automatica, 2002, 38(2), 261–267.

Walsh, G., Ye, H. & Bushnell, L. (2002b). Stability analysis of networked control systems, IEEE Transactions on Control Systems Technology, 10(3), 428-446.

Wang F.Y. and Liu D. (2008). Networked control systems: theory and applications, Springer-Verlag, London, p.2.

Wang, Z.-Q., Lundstrom, P., & Skogestad, S., (1994). Representation of uncertain time delays in the H infinity framework, International Journal of Control, 59(3), 627-638.

Willems, J.C. (1986). From time series to linear system - Part 1. Finite dimensional linear time invariant systems, Automatica, 22(5), 561-580.

Wittenmark, B., Nilsson, J. & Törngren, M. (1995). Timing problems in real-time control systems, In Proceeding of the American Control Conference, (ACC95), Seattle, 2000-2004.

Yang, T. C. (2006). Networked control system: a brief survey, IEE Proceedings Control Theory and Applications, 153(4), 403-412.

Yi, J., Wang, Q. Zhao, D. & Wen, J.T. (2007). BP neural network prediction-based variable-period sampling approach for networked control systems, Applied Mathematics and Computation, 185(2), 976–988.

Ylinen, R. Separation theorem for polynomial feedback systems. 3rd IFAC Robust Control Design (ROCOND 2000). Prague, Czech Republic. 6 pp., 2000.

Ylinen, R. (1975). On the algebraic theory of linear differential and difference systems with timevarying or operator coefficients. Helsinki University of Technology, Systems Theory Laboratory, B23.

Ylinen, R. (1980) An algebraic theory for analysis and synthesis of time-varying linear differential systems, Acta Polytechnica Scandiavica, Ma32.

Ylinen, R. & N. Vatanski. (2007) Control of systems with time-varying delays. Networked Systems Tolerant to Faults (NeCST) workshop, Nancy 2007.

Yook, J.K., Tilbury, D.M. & Soparkar, N.R. (2002). Trading computation for bandwidth: reducing communication in distributed control systems using state estimators, IEEE Transactions on control systems technology, 10(4), 503-518.

Yu, M., Wang, L. & Chu, T. (2005). Sampled-data stabilisation of networked control systems with nonlinearity, IEE Proceeding Control Theory and Applications, 152(6), 609-614.

Yue, D., Han, Q.L. & Lam, J. (2005). Network-based robust H_∞ control of systems with uncertainty, Automatica, 41(6), 999 – 1007.

Yue, D. & Han, Q.L. (2005). Delayed feedback control of uncertain systems with time-varying input delay, Automatica, 41(2), 233 – 240.

Zhang, L., & Hua-Jing, F. (2006). A novel controller design and evaluation for networked control systems with time-variant delays, Journal of the Franklin Institute, 343(2), 161–167.

Zhang, L., Shi, Y., Chen, T., & Huang, B. (2005). A new method for stabilization of networked control systems with random delays, IEEE Transactions on Automatic Control, 50(8), 1177–1181.

Zhang, H.S., Xie, L.H., Wang,W. & Lu, X. (2004). An innovation approach to H-infinity fixed-lag smoothing for continuous time-varying systems, IEEE Transactions on Automatic Control, 49(12), 2240–2244.

Aalto-DD 100/2011

ISBN 978-952-60-4325-8 (pdf)
ISBN 978-952-60-4324-1
ISSN-L 1799-4934
ISSN 1799-4942 (pdf)
ISSN 1799-4934

Aalto University
School of Chemical Technology
Department of Biotechnology and Chemical Technology
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS